(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TRANSISTOR SIZING IN THE DESIGN OF
HIGH-SPEED CMOS SUPER BUFFERS

by

Gordon R. Steele

March 1988

Co-advisor: D.E. Kirk
Co-advisor: H.H. Loomis, Jr.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE A10 591

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION (UNCLASSIFIED)	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

TRANSISTOR SIZING IN THE DESIGN OF HIGH-SPEED CMOS SUPER BUFFERS (U)

12. PERSONAL AUTHOR(S)
Steele, Gordon R.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM _____ TO _____	1988 March	145

16. SUPPLEMENTARY NOTATION The views in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	MacPITTS; Silicon Compiler; CMOS; VLSI; Super Buffer; Transistor Sizing; High-Speed CMOS

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

An algorithm for sizing transistors for static Complementary-symmetry Metal-Oxide-Semiconductor (CMOS) integrated circuit logic design using silicon gate enhancement mode Field-Effect Transistors (FET) is derived and implemented in software. The algorithm is applied to the mask level hardware design of a three micron minimum feature size p well high-speed super buffer. A software representation of the super buffer can be used for the automated design of custom Very-Large-Scale Integrated (VLSI) circuits.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION (UNCLASSIFIED)
22a. NAME OF RESPONSIBLE INDIVIDUAL H.H. Loomis, Jr.	22b. TELEPHONE (Include Area Code) (408) 646-3214 22c. OFFICE SYMBOL 62LM

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—606-24.

UNCLASSIFIED

Approved for public release; distribution is unlimited

Transistor Sizing in the Design of
High-Speed CMOS Super Buffers

by

Gordon R. Steele
Captain, United States Marine Corps
B.S.F.S., Georgetown University, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

March 1988

Author: _____
Gordon R. Steele

Approved by: _____
Donald E. Kirk, Co-Advisor

_____
Herschel H. Loomis Jr., Co-Advisor

_____
John P. Powers, Chairman
Department of Electrical and Computer Engineering

_____
Gordon E. Schacher, Dean of Science and Engineering

ABSTRACT

An algorithm for sizing transistors for static Complementary-symmetry Metal-Oxide-Semiconductor (CMOS) integrated circuit logic design using silicon gate enhancement mode Field-Effect Transistors (FET) is derived and implemented in software. The algorithm is applied to the mask level hardware design of a three micron minimum feature size p well high-speed super buffer. A software representation of the super buffer can be used for the automated design of custom Very-Large-Scale-Integrated (VLSI) circuits.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

My deepest and most sincere thanks for the support I have received in the pursuit of my studies go to my wife, Anne, and my children, Andrew and Julia. I would also like to express my appreciation to Dr. Donald Kirk and to Dr. Herschel Loomis of the Naval Postgraduate School for the assistance and guidance that they provided to me in conducting the research and the writing of this thesis.

I. INTRODUCTION

The MacPITTS silicon compiler is a tool for the automated design of complete, Very-Large-Scale Integrated (VLSI) microchips that was created at the Massachusetts Institute of Technology Lincoln Laboratory in the early 1980's [Massachusetts Institute of Technology 1982 Conference on Advanced Research in VLSI, 1982, pp. 28–29]. To use MacPITTS a design engineer specifies the desired logic behavior of a complete microchip in a source program written in a high level algorithmic language. Writing this program is the only task that the design engineer need perform—the entire design process is then performed by the compiler. The compiler extracts information from this program which it uses to create a mask level layout in Caltech Intermediate Form (CIF). The layout when specified in this form is suitable for use by silicon foundries for the fabrication of an actual microchip.

Use of a silicon compiler provides dramatic design time savings over handcrafting layouts. The design—simulate—redesign cycle that occurs before fabrication of handcrafted layouts is lengthy and therefore expensive. Handcrafted designs that take many months of man-hours to complete using computer based engineering design tools may be completed in a much shorter time by using a silicon compiler. Most VLSI designs today contain 20,000 to 50,000 devices. When human engineers design these large chips they partition and design the project in teams. Integrating the subsystems into a final working design requires close coordination throughout the design process and a uniform approach on the part of all the design teams. Using a silicon compiler to design automatically institutes uniformity in the design process. The compiler uses embedded techniques and rigorous checking

and verification of design practices to ensure that the resulting circuit is "correct by construction." These embedded functions often incorporate the practices of expert integrated circuit designers. A good compiler can verify the functionality and timing of a design after it has been implemented. Some can provide real-time checking, verification and implementation strategies that flag potential errors and hazards and prevent them from being entered into the design. All of these features of silicon compilers contribute to the production of designs that are as correct as possible before fabrication. This is extremely important in VLSI design because it is difficult and costly to isolate design errors on a finished microchip. [Pollack, Erickson, and Mazor, 1986, pp. 79–80]

The MacPITTS Silicon Compiler is actually a collection of programs each of which performs a specific function. The base language of these programs is Franz Lisp with extended functions for geometric cell layout provided by the Massachusetts Institute of Technology Lincoln Laboratory language L5 and by MacPITTS itself. One way in which the programs can be categorized is to lump together all those that are dependent upon a particular microchip fabrication technology and those that are not. By altering only the technology dependent programs to reflect the design rules of a desired technology the rest of the compiler can become technology independent. This is a very desirable feature in the fast moving electronics industry because a favored technology today may be outmoded tomorrow. The technology dependent programs in the original MacPITTS compiler were based on n channel Metal-Oxide-Semiconductor (NMOS) technology. NMOS has since been largely replaced by Complementary-symmetry Metal-Oxide-Semiconductor (CMOS) technology as the commercial technology of choice for low power consumption VLSI applications. This transition in industry has occurred largely because

CMOS technology has matured to the point that it now rivals NMOS in switching speed yet consumes only a fraction of the power required by NMOS circuitry. It is therefore a logical step to modify the technology dependent portions of the MacPITTS Silicon Compiler to support CMOS design rules.

One of the major technology dependent portions of MacPITTS is the body of code within it that defines the geometric layout of logic cells. The compiler selects, places, and interconnects these cells in such a way that the result is a design for a physically achievable microchip that will behave as specified by the word picture that the design engineer presents to the compiler in his or her source program. These cells reside in what is essentially a library.

An ongoing project at the Naval Postgraduate School in Monterey, California, has been the addition of a CMOS option to the technology dependent portions of the MacPITTS Silicon Compiler. Since the work has progressed to the point that the present compiler is a significant departure from the original MacPITTS the hybrid NMOS/CMOS compiler is now referred to as the Monterey Silicon Compiler.

The immediate goal of the research documented in this thesis is to create and insert into the Monterey Silicon Compiler code a mask level integrated circuit design of a high-speed static CMOS super buffer to replace the existing NMOS version. In the Monterey Silicon Compiler architecture the super buffer is that part of the branching fan-out tree that feeds clock signals from off-chip to various parts of a VLSI circuit. Since the super buffer is a gate in the path of incoming clock signals it must be guaranteed to function within specified timing constraints under given loading lest the operation of an entire chip fail due to the inability of the clock circuit to drive the on chip storage registers within an acceptable amount of time delay. From the designer's point of view the timing characteristics

of enhancement Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFET), of which the super buffer is made, are controlled chiefly by the mask geometry of the MOSFETs. Therefore, a major portion of this research is devoted to an analysis of transistor sizing for static CMOS logic gates constructed from silicon gate enhancement MOSFETs.

Transistor sizing is the principal means by which an integrated circuit design engineer can influence the timing of a MOSFET. In a silicon compiler the design engineer is a computer. Typically, human engineers achieve desired timing through a design—simulate—redesign iterative method based on rules of thumb that are handed down by successful engineers to junior engineers. A silicon compiler that strives to optimize signal input to signal output delay time on a microchip cannot work in this way—it must be given explicit instructions on how to decide what transistor sizes are required to drive a given load, with a given supply voltage and operating temperature, within a specified delay time. Chapter II, which deals with the transistor sizing problem, is an original contribution to the field of time delay optimization in silicon compiler generated custom CMOS VLSI circuits. A coherent body of mathematical equations that detail how to size transistors to achieve desired timing parameters under known loading conditions, operating temperature, and supply voltage is derived in Chapter II. The equations are applicable to general static CMOS design.

Although the Monterey Silicon Compiler is not capable of designing a circuit that is constructed specifically to achieve a designer specified time delay this is a topic applicable to the general field of silicon compilation. The equations in Chapter II are useful to the Monterey Silicon Compiler as a design tool for the construction of the individual logic cells in the library that the compiler uses to build an integrated circuit floorplan. The equations developed are implemented in

a computer program that is included in Appendix A. The program is used to apply the equations to a design problem, the design of the super buffer, in Chapter III. The resulting mask level design is simulated with a SPICE MOSFET model that is independent of the transistor sizing equations to verify the correctness of the approach. The design is translated into L5 code and inserted into the Monterey Silicon Compiler code for use in the automated design of custom CMOS VLSI circuits. The design of the super buffer moves the Monterey Silicon Compiler project one step closer to completion.

II. SIZING OF SILICON GATE ENHANCEMENT MODE METAL-OXIDE-SEMICONDUCTOR FIELD-EFFECT TRANSISTORS IN STATIC CMOS LOGIC DESIGN

A. CHARACTERISTICS OF STATIC CMOS CIRCUIT DESIGN

Static CMOS Logic gates are made from two distinct blocks of devices that are connected together. Within one block are series combinations of p MOSFETs or parallel combinations of p MOSFETs or both. The MOSFETs in the p block are connected to a supply voltage.

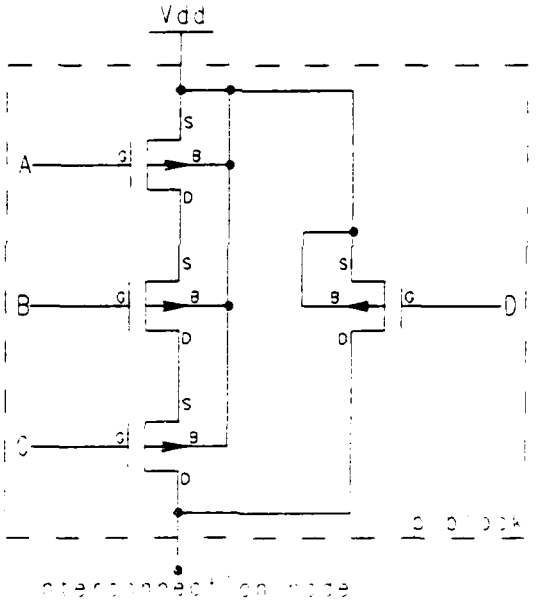


Figure 2.1 Schematic of a Possible p Block Configuration in a Static CMOS Logic Gate

Figure 2.1 shows an example of a possible p block configuration [Weste and Eshraghian, 1985, p. 16]. Within the other block are series combinations of n MOSFETs or parallel combinations of n MOSFETs or both. The MOSFETs in

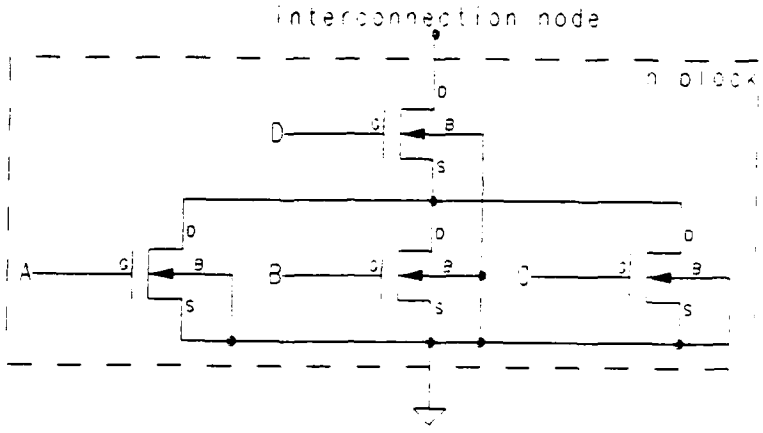


Figure 2.2 Schematic of a Possible n Block Configuration in a Static CMOS Logic Gate

the n block are connected to ground. Figure 2.2 shows an example of a possible n block configuration. [Weste and Eshraghian, 1985, p. 16]

Any load capacitance to be driven by a static CMOS logic gate is attached to the node defined by the interconnection of the p and n blocks. Figure 2.3 shows the interconnection of the two blocks with a capacitive load attached.

The simplest static CMOS logic gate is an inverter. The inverter shown in Figure 2.4 consists of a single p MOSFET and a single n MOSFET. In an ideal sense, each p MOSFET in a static CMOS logic gate may be thought of as a switch along a path connecting the supply voltage to the load capacitance [Weste and Eshraghian, 1985, p. 7]. Similarly, each n MOSFET may be thought of as a switch along a path connecting the load capacitance to ground.

Figure 2.5 illustrates that a MOSFET that is conducting can be thought of as a closed switch and that a MOSFET that is in cutoff can be thought of as an open switch. Therefore, the voltage across the terminals of the load capacitance

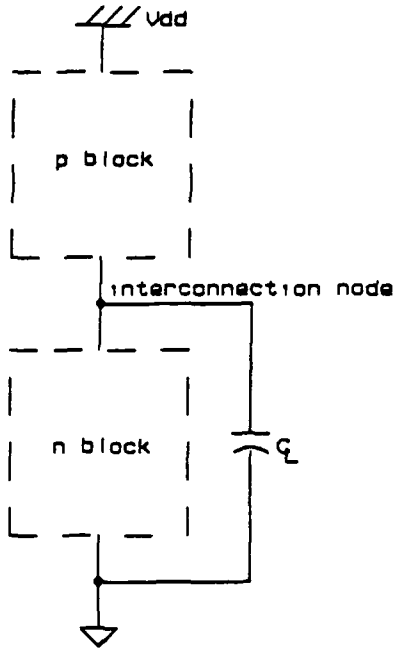


Figure 2.3 Interconnection of p and n Blocks in Static CMOS Logic Gates with Capacitive Loading

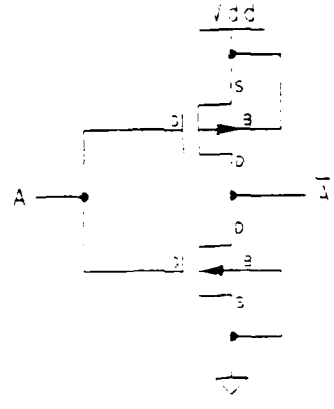


Figure 2.4 Schematic of a Static CMOS Inverter without Capacitive Loading

will be driven to the logic high level if enough switches in the p block are closed to create at least one complete path from the supply voltage to the load capacitance and enough switches in the n block are opened to sever all paths between the load capacitance and ground. Figure 2.6.a illustrates this switch level behavior for the inverter in Figure 2.4. The voltage across the terminals of the load capacitance will be driven to the logic low level if enough switches in the n block are closed to create at least one path from the load capacitance to ground and enough switches in the p block are opened to sever all paths between the supply voltage and the load. Figure 2.6.b illustrates this switch level behavior for the inverter in Figure 2.4.

B. DERIVATION OF EQUATIONS FOR TRANSISTOR SIZING IN A STATIC CMOS INVERTER

1. Overview

In the physical world MOSFETs are not ideal switches. In addition to the capacitive load that they must drive they have internal parasitic capacitances and resistances that must also be driven. These parasitic capacitances and resistances are chiefly determined by operating temperature, various parameters associated with the fabrication process, and the physical dimensions of the MOSFETs. Only the third factor, the physical dimensions of the MOSFETs, is under the control of a circuit designer.

It is essential that the circuit designer, be it a human or an automated CAD tool like a silicon compiler, have a method to determine the physical dimensions of the MOSFETs in a static CMOS logic gate that are required to drive the load. The method presented here finds the dimensions of the p and n MOSFETs in the inverter of Figure 2.4 that are required to drive the parasitics and known capacitive load within a designer specified time constraint. The inverter is used

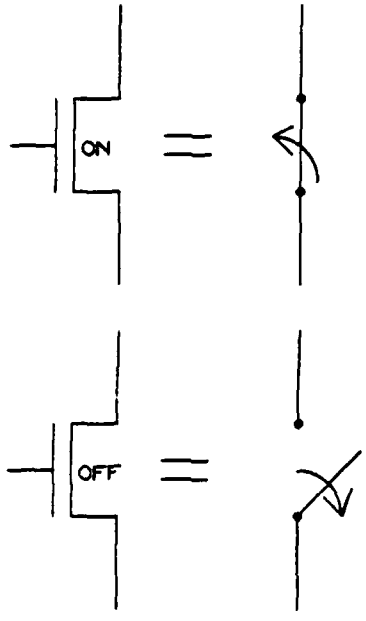
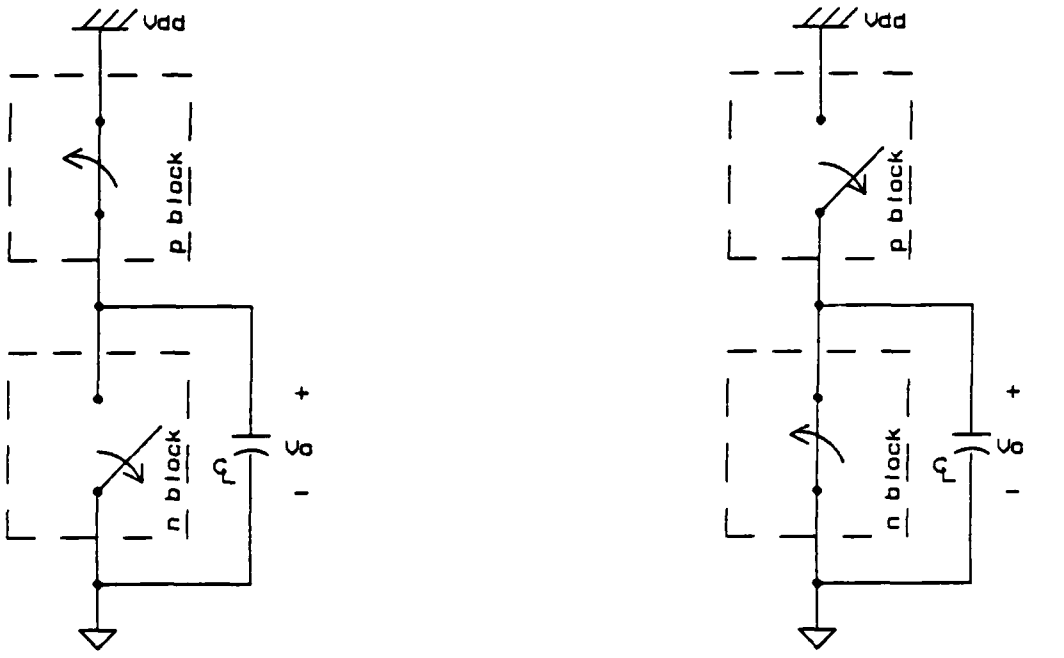


Figure 2.5 Switch Level Idealization of MOSFET Behavior

for the development of this method because it is the simplest static CMOS logic gate and as such the method developed for the inverter can be easily extended to general static CMOS logic design, the topic of Section C.

2. Rise Time and Fall Time Determination

Table 2.1 gives the ideal (first order) equations of the Shichman-Hodges model describing the behavior of MOSFETs in their three regions of operation: cutoff region, linear region, and saturation region [Hodges and Jackson, 1983, p. 51, and MicroSim Corp., 1987, p. 78]. Table 2.1 refers to the variable $KP\{T\}$. KP is one of the 58 variables used in the SPICE circuit simulation program, a standard CAD tool for the design and simulation of MOSFET circuits. Table 2.2 presents only those MOSFET model parameters used in this thesis as defined in MicroSim Corporation's PSpice program documentation [MicroSim Corp., 1987,



(a) Inverter Driving V_o to the Logic High Level.

(b) Inverter Driving V_o to the Logic Low Level.

Figure 2.6 Switch Level Idealization of Static CMOS Inverter Behavior

pp. 74–76]. The model parameters can be grouped into two broad categories: those that represent the mask geometry of the on-chip devices and those that represent the electrical and metallurgical properties of the chip due to the particular type of fabrication process used to construct the microchip.[1] As stated in Section B.1,

[1] A good discussion of the physical meaning of the SPICE MOSFET model parameters can be found in, "The Simulation of MOS Integrated Circuits Using SPICE2, Memorandum No. M80/7, February, 1980", by Vladimirescu and Liu. The paper is available through the Electronics Research Laboratory Industrial Support Office at the University of California at Berkeley.

the circuit designer is able to modify parameters in the first category but not those in the second.

Some of these parameters are temperature dependent. The effects of temperature on those SPICE MOSFET model parameters used in this thesis are defined by the equations in Table 2.3. These equations are drawn from the program documentation for PSpice by MicroSim Corporation. [MicroSim Corp., 1987, p. 80]

Note that positive drain current in an n channel enhancement MOSFET flows from drain to source. This is illustrated in Figure 2.7.a. In a p channel enhancement MOSFET it flows from source to drain [Sedra and Smith, 1982, p. 311]. This is illustrated in Figure 2.7.b.

Section B.1 states that the load on a MOSFET is capacitive. Switching from one logic level to another therefore involves the charging or discharging of the parasitic and load capacitors and this requires time. The time that it takes a voltage waveform to rise from 10% to 90% of its steady state value is defined to be its rise time, denoted t_r. Fall time, denoted t_f, is defined to be the time it takes a voltage waveform to fall from 90% to 10% of its steady state value. [Weste and Eshraghian, 1985, p. 137]

The following discussion is based on a method of finding the rise and fall time of a static CMOS inverter proposed by Weste and Eshraghian [Weste and Eshraghian, 1985, pp. 137–140]. Given the circuit of Figure 2.8 in which C_L is initially charged so that $V_0 = Vdd$, a pulse of magnitude Vdd is applied to Vin. The effect of the pulse on the circuit's behavior is plotted in Figure 2.9.a. In this figure the regions of operation of the n MOSFET and p MOSFET are labeled in accordance with the definitions of the regions presented in Table 2.1. From Figure 2.9.a it is evident that the p MOSFET immediately enters the cutoff region where

TABLE 2.1 MOSFET DRAIN CURRENT EQUATIONS (FROM MICROSIM CORPORATION'S PSPICE)

	DEVICE TYPE	REGION	DRAIN CURRENT
	n	$Vds_n \geq 0$ $Vgs_n - Vto_n\{T\} < 0$ [cutoff region]	$I_{\text{drain}_n} = 0$
*	n	$Vds_n \geq 0$ $Vds_n < Vgs_n - Vto_n\{T\}$ [linear region]	$I_{\text{drain}_n} = \left(\frac{W_n}{L_{eff_n}}\right) \cdot \left(\frac{KP_n\{T\}}{2}\right) \cdot Vds_n$ $\cdot\left(2 \cdot (Vgs_n - Vto_n\{T\}) - Vds_n\right)$
*	n	$Vds_n \geq 0$ $0 \leq Vgs_n - Vto_n\{T\} \leq Vds_n$ [saturation region]	$I_{\text{drain}_n} = \left(\frac{W_n}{L_{eff_n}}\right) \cdot \left(\frac{KP_n\{T\}}{2}\right)$ $\cdot(Vgs_n - Vto_n\{T\})^2$
	p	$\lvert Vds_p\rvert \geq 0$ $\lvert Vgs_p\rvert - \lvert Vto_p\{T\}\rvert < 0$ [cutoff region]	$I_{\text{drain}_p} = 0$
†	p	$\lvert Vds_p\rvert \geq 0$ $\lvert Vds_p\rvert < \lvert Vgs_p\rvert - \lvert Vto_p\{T\}\rvert$ [linear region]	$I_{\text{drain}_p} = \left(\frac{W_p}{L_{eff_p}}\right) \cdot \left(\frac{KP_p\{T\}}{2}\right) \cdot \lvert Vds_p\rvert$ $\cdot\left(2 \cdot (\lvert Vgs_p\rvert - \lvert Vto_p\{T\}\rvert) - \lvert Vds_p\rvert\right)$
†	p	$\lvert Vds_p\rvert \geq 0$ $0 \leq \lvert Vgs_p\rvert - \lvert Vto_p\{T\}\rvert \leq \lvert Vds_p\rvert$ [saturation region]	$I_{\text{drain}_p} = \left(\frac{W_p}{L_{eff_p}}\right) \cdot \left(\frac{KP_p\{T\}}{2}\right)$ $\cdot(\lvert Vgs_p\rvert - \lvert Vto_p\{T\}\rvert)^2$

where:

$$Vto_n\{T\} = VTO_n + GAMMA_n \cdot \left((PHI_n\{T\} + |Vbs_n|)^{1/2} - (PHI_n\{T\})^{1/2}\right);\ Vbs_n \leq 0$$

$$|Vto_p\{T\}| = |VTO_p| + GAMMA_p \cdot \left((PHI_p\{T\} + Vbs_p)^{1/2} - (PHI_p\{T\})^{1/2}\right);\ Vbs_p \geq 0$$

* $L_{eff_n} = L_n - 2 \cdot LD_n$

† $L_{eff_p} = L_p - 2 \cdot LD_p$

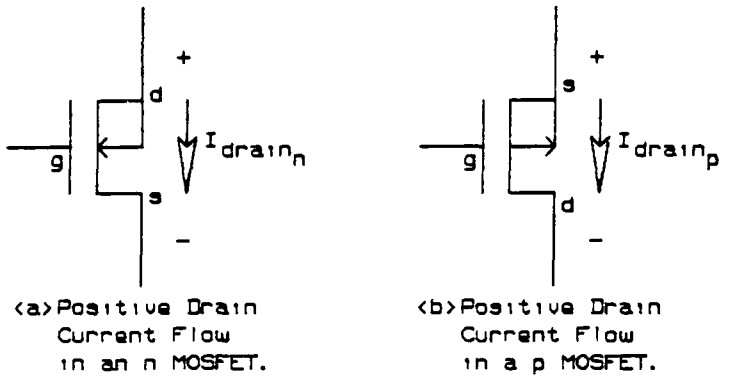


Figure 2.7 Current Flow in MOSFETs

$I_{\text{drain}_p} = 0$. The n MOSFET passes through two regions as V_0 falls: the saturation region, then the linear region. It is possible then to redraw Figure 2.8 as Figure 2.9.b during fall time with the understanding that I_{drain_n} is dependent upon the region of operation of the n MOSFET as the capacitor voltage, V_0, falls. Indeed, this is the reason for the use of a dependent current source symbol in Figure 2.9.b. From Weste and Eshraghian [Weste and Eshraghian, 1985, p. 139]:

"... it is evident that the fall time, t_f, consists of two intervals:

1. t_{f_1} = period during which the capacitor voltage, V_0, drops from $0.9 \cdot Vdd$ to $(Vdd - Vto_n\{T\})$ $[I_{\text{drain}_n} \equiv$ saturation current$]$.

2. t_{f_2} = period during which the capacitor voltage, V_0, drops from $(Vdd - Vto_n\{T\})$ to $0.1 \cdot Vdd$ $[I_{\text{drain}_n} \equiv$ linear current$]$."

Figure 2.9.b can now be used to write an equation at the interconnection node based on Kirchoff's current law: [2]

$$I_{\text{drain}_n} - (-i_c) = 0$$

[2] Current flow convention: Current flowing out of a node is positive, current flowing into a node is negative.

TABLE 2.2 PSPICE MOSFET MODEL PARAMETERS (FROM MICROSIM CORPORATION'S PSPICE)

L	channel length	meter
W	channel width	meter
AD	drain diffusion area	square meter
AS	source diffusion area	square meter
PD	drain diffusion perimeter	meter
PS	source diffusion perimeter	meter
LEVEL	model type (1, 2, or 3)	
LD	lateral diffusion	meter
VTO	0-bias threshold voltage	volt
KP	transconductance	amp/volt^2
GAMMA	bulk threshold parameter	volt$^{1/2}$
PHI	surface potential	volt
LAMBDA	channel-length modulation	volt^{-1}
RD	drain ohmic resistance	ohm
RS	source ohmic resistance	ohm
RSH	drain, source diffusion sheet resistance	ohm/sq.
PB	bulk junction potential	volt
CJ	bulk junction 0-bias bottom capacitance/area	farad/meter^2
CJSW	bulk junction 0-bias perimeter capacitance/length	farad/meter
MJ	bulk junction bottom grading	
MJSW	bulk junction sidewall grading	
FC	bulk junction fwd-bias cap. coef	
CGSO	G-S overlap cap./channel width	farad/meter
CGDO	G-D overlap cap./channel width	farad/meter
NSUB	substrate doping density	1/cm^3
NSS	surface state density	1/cm^2
NFS	fast surface state density	1/cm^2
TOX	oxide thickness	meter
TPG	gate material: +1 = opposite of substrate −1 = same as substrate 0 = aluminum	
XJ	metallurgical junction depth	meter
UO	surface mobility	cm^2/volt-sec
UCRIT	mobility degradation critical field	volt/cm
UEXP	mobility degradation exponent	
VMAX	max. drift velocity	meter/sec
NEFF	channel charge coef.	
DELTA	width effect on threshold	

TABLE 2.3 MOSFET TEMPERATURE EFFECTS (FROM MICROSIM CORPOPATION'S PSPICE)

$$EG\{T\} = 1.16 - (0.000702 \cdot T^2) \,/\, (T + 1108)$$

$$PB\{T\} = PB \cdot T/Tnom - 3 \cdot Vt \cdot ln(T/Tnom) - EG\{T\} + Eg\{Tnom\} \cdot T/Tnom$$

$$PHI\{T\} = PHI \cdot T/Tnom - 3 \cdot Vt \cdot ln(T/Tnom) - EG\{T\} + Eg\{Tnom\} \cdot T/Tnom$$

$$CJ\{T\} = CJ \cdot \Big(1 + MJ \cdot \big(.0004 \cdot (T - Tnom) + (1 - PB\{T\}/PB)\big)\Big)$$

$$CJSW\{T\} = CJSW \cdot \Big(1 + MJSW \cdot \big(.0004 \cdot (T - Tnom) + (1 - PB\{T\}/PB)\big)\Big)$$

$$KP\{T\} = KP \cdot (T/Tnom)^{-3/2}$$

$$UO\{T\} = UO \cdot (T/Tnom)^{-3/2}$$

$$\epsilon_{rel,oxide}\{T\} = (KP\{T\} \cdot 100 \cdot TOX)/(UO\{T\} \cdot \epsilon_0)$$

Note:

All temperatures in degrees Kelvin

$Tnom$ = nominal temperature (assumed to be 300.15 K)

ϵ_0 = Electric Field Constant (permittivity of vacuum) $\left[\frac{F}{cm}\right]$

$Vt = K \cdot T/q$ (thermal voltage)

K = Boltzmann's constant

q = electron charge

T = operating temperature

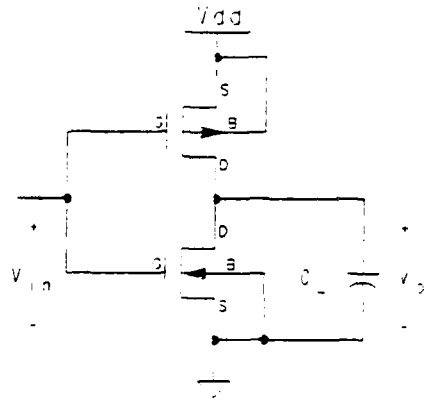


Figure 2.8 Schematic of a Static CMOS Inverter with Capacitive Loading

$$I_{\mathbf{drain_n}} + C_L \cdot \frac{dV_0}{dt} = 0 \qquad ; \quad i_c = C_L \cdot \frac{dV_0}{dt} \tag{2.1}$$

Applying equation (2.1) to the first interval where $I_{\mathbf{drain_n}}$ = saturation current gives:

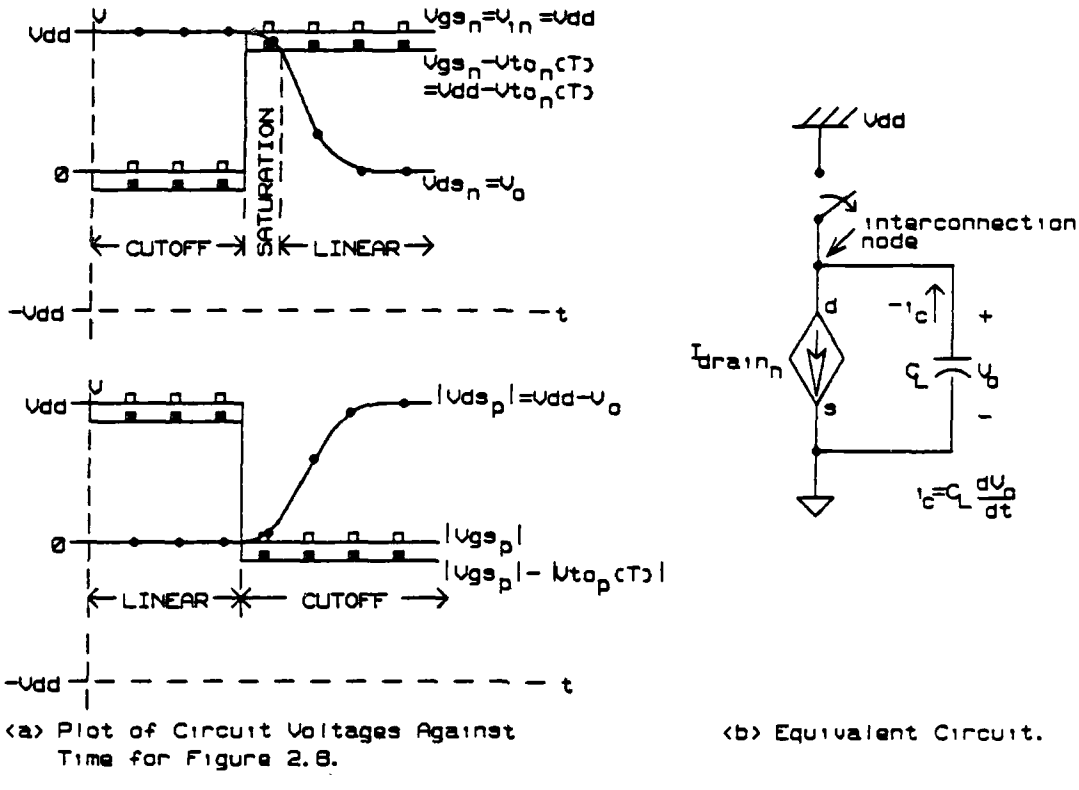
$$\left(\frac{W_n}{L_{effn}}\right) \cdot \left(\frac{KP_n\{T\}}{2}\right) \cdot \left(Vgs_n - Vto_n\{T\}\right)^2 + C_L \cdot \frac{dV_0}{dt} = 0$$

After substituting $Vds_n = V_0$ and $Vgs_n = Vin = Vdd$ and some algebraic manipulation this equation becomes:

$$dt = -\frac{2 \cdot C_L}{\left(\frac{W_n}{L_{effn}}\right) \cdot \left(KP_n\{T\}\right) \cdot \left(Vdd - Vto_n\{T\}\right)^2} \cdot dV_0 \tag{2.2}$$

Integrating the left side of equation (2.2) from $t = t_1$ to $t = t_2$ and the right side over the corresponding output voltages $V_0 = 0.9Vdd$ to $V_0 = Vdd - Vto_n\{T\}$ gives:

$$t_{f_1} = \frac{2 \cdot C_L \cdot \left(Vto_n\{T\} - 0.1 \cdot Vdd\right)}{\left(\frac{W_n}{L_{effn}}\right) \cdot \left(KP_n\{T\}\right) \cdot \left(Vdd - Vto_n\{T\}\right)^2} \tag{2.3}$$



(a) Plot of Circuit Voltages Against Time for Figure 2.8.

(b) Equivalent Circuit.

Figure 2.9 Effects of the Rising Edge of a Pulsed Input on a Static CMOS Inverter

Equation (2.1) may be applied to the second interval where $I_{\mathbf{drain_n}}$ = linear current:

$$\left(\frac{W_n}{L_{effn}}\right)\cdot\left(\frac{KP_n\{T\}}{2}\right)\cdot Vds_n\cdot(2\cdot(Vgs_n-Vto_n\{T\})-Vds_n)+C_L\cdot\frac{dV_0}{dt}=0 \quad (2.4)$$

Again substituting $Vds_n = V_0$ and $Vgs_n = V_{in} = Vdd$ and performing some algebra gives:

$$dt=\frac{2\cdot C_L}{\left(\frac{W_n}{L_{effn}}\right)\cdot(KP_n\{T\})\cdot(V_0^2-2\cdot V_0)}\cdot dV_0 \quad (2.5)$$

Integrating the left side of equation (2.5) from $t = t_2$ to $t = t_3$ and the right side over the corresponding output voltages $V_0 = Vdd - Vto_n\{T\}$ to $V_0 = 0.1Vdd$ gives:

$$t_{f_2} = \frac{C_L}{\left(\frac{W_n}{L_{effn}}\right) \cdot (KP_n\{T\}) \cdot (Vdd - Vto_n\{T\})} \cdot \log_e \left(\frac{19 \cdot Vdd - 20 \cdot Vto_n\{T\}}{Vdd}\right) \tag{2.6}$$

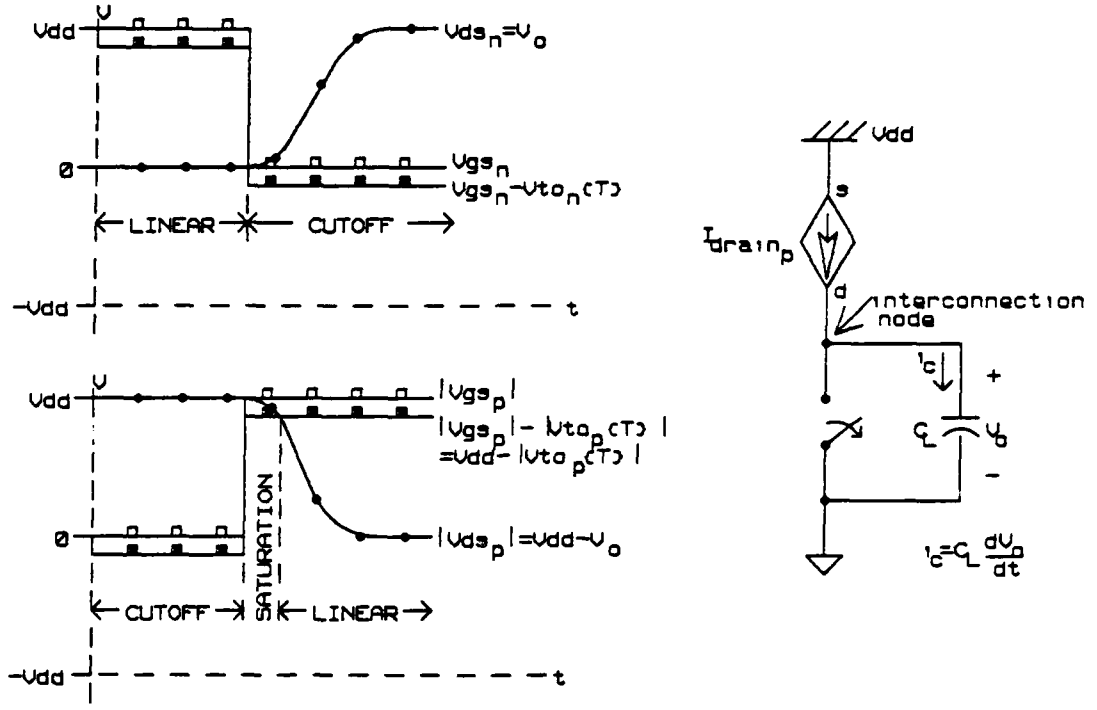
The total fall time, t_f, is found by summing equations (2.3) and (2.6):

$$t_f = t_{f_1} + t_{f_2} = \frac{2 \cdot C_L}{\left(\frac{W_n}{L_{effn}}\right) \cdot (KP_n\{T\}) \cdot (Vdd - Vto_n\{T\})}$$

$$\cdot \left[\frac{(Vto_n\{T\} - 0.1 \cdot Vdd)}{(Vdd - Vto_n\{T\})} + \frac{1}{2} \cdot \log_e \left(\frac{19 \cdot Vdd - 20 \cdot Vto_n\{T\}}{Vdd}\right)\right] \tag{2.7}$$

A similar procedure may be applied to find the rise time, t_r. Given the circuit of Figure 2.8 in which C_L is initially discharged so that $V_0 = 0$, the input voltage instantaneously falls from $V_{in} = Vdd$ to $V_{in} = 0$. The effect of the input voltage on the circuit's behavior is plotted in Figure 2.10.a. In this figure the regions of operation of the n and p MOSFETs are labeled in accordance with the definitions of the regions presented in Table 2.1. From the figure it is evident that the n MOSFET immediately enters the cutoff region where $I_{\text{drain}_n} = 0$. The p MOSFET passes through two regions, as V_0 rises: the saturation region then the linear region. It is possible then to redraw Figure 2.8 as Figure 2.10.b during rise time with the understanding that I_{drain_p} is dependent upon the region of operation of the p MOSFET as the capacitor voltage, V_0, rises. Figure 2.10.b can be used to write an equation at the interconnection node based on Kirchoff's Current law:

$$-I_{\text{drain}_p} + i_c = 0 \tag{2.8}$$



(a) Plot of Circuit Voltages Against Time for Figure 2.8.

(b) Equivalant Circuit.

Figure 2.10 Effects of the Falling Edge of a Pulsed Input on a Static CMOS Inverter

Since the p MOSFET passes through two regions of operation during the rise time, equation (2.8) must be applied separately to each region and the separate results are summed together. The lower voltage limit in the time span denoted t_{r_1} is $0.1Vdd$. The upper limit may be found from Figure 2.10.a. In Figure 2.10.a. the p MOSFET crosses over from the saturation region to the linear region when:

$$|Vds_p| = |Vgs_p| - |Vto_p\{T\}| \tag{2.9}$$

But

$$|Vds_p| = Vdd - V_0 \tag{2.10}$$

$$|Vgs_p| - |Vto_p\{T\}| = Vdd - |Vto_p\{T\}| \tag{2.11}$$

Substituting equations (2.10) and (2.11) into (2.9) and solving for V_0 gives:

$$V_0 = |Vto_p\{T\}| \tag{2.12}$$

Thus the rise time intervals t_{r_1} and t_{r_2} are defined:

1. t_{r_1} = period during which the capacitor voltage, V_0, rises from $0.1Vdd$ to $|Vto_p\{T\}|$ $\left[I_{\text{drain}_p} \equiv \text{saturation current}\right]$.

2. t_{r_2} = period during which the capacitor voltage, V_0, rises from $|Vto_p\{T\}|$ to $0.9Vdd$ $\left[I_{\text{drain}_p} \equiv \text{linear current}\right]$.

The drain current equations are substituted into equation (2.8) for the two regions of operation of the p MOSFET in identical fashion to the procedure used in determining t_f for the n MOSFET. The result is:

$$t_r = t_{r_1} + t_{r_2} = \frac{2 \cdot C_L}{\left(\frac{W_p}{L_{eff_p}}\right) \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)}$$

$$\cdot \left[\frac{(|Vto_p\{T\}| - 0.1 \cdot Vdd)}{(Vdd - |Vto_p\{T\}|)} + \frac{1}{2} \cdot \log_e \left(\frac{19 \cdot Vdd - 20 \cdot |Vto_p\{T\}|}{Vdd} \right) \right] \tag{2.13}$$

Figure 2.11 shows an idealized MOSFET where L and W are the length and width of the channel, respectively. In the fabrication process that produces

this ideal MOSFET doping ions diffuse only vertically down into the substrate through designer specified windows in the mask that define the source and drain regions. Figure 2.12 shows the result of a more true to life fabrication process. In this process the doping ions diffuse not only vertically down into the substrate but laterally under the edges of the window as well. The effect is that the drain and source regions are bloated in all directions by an amount LD. Since this lateral diffusion also occurs at both ends of the channel, the channel length is effectively reduced by $2 \cdot LD$. The effective channel length is denoted L_{eff} in Table 2.1 and in equations (2.7) and (2.13). For short channel devices this decrease in channel length can have a significant impact on drain current.

Rewriting equations (2.7) and (2.13) in terms of effective channel length gives:

$$t_f = \frac{2 \cdot C_L}{\left(\frac{W_n}{L_n - 2 \cdot LD_n}\right) \cdot (KP_n\{T\}) \cdot (Vdd - Vto_n\{T\})} \cdot \left[\frac{(Vto_n\{T\} - 0.1 \cdot Vdd)}{(Vdd - Vto_n\{T\})} + \frac{1}{2} \cdot \log_e\left(\frac{19 \cdot Vdd - 20 \cdot Vto_n\{T\}}{Vdd}\right)\right] \quad (2.14)$$

$$t_r = \frac{2 \cdot C_L}{\left(\frac{W_p}{L_p - 2 \cdot LD_p}\right) \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)} \cdot \left[\frac{(|Vto_p\{T\}| - 0.1 \cdot Vdd)}{(Vdd - |Vto_p\{T\}|)} + \frac{1}{2} \cdot \log_e\left(\frac{19 \cdot Vdd - 20 \cdot |Vto_p\{T\}|}{Vdd}\right)\right] \quad (2.15)$$

where $Vto_n\{T\}$ and $|Vto_p\{T\}|$ are defined in Table 2.1 and $KP_n\{T\}$ and $KP_p\{T\}$ are defined in Table 2.3.

Delay time, denoted t_d, is the time it takes a logic transition to pass from the input of a gate to its output. In fully restored CMOS logic t_d is explicitly defined to be the amount of time that elapses between the moment that the input

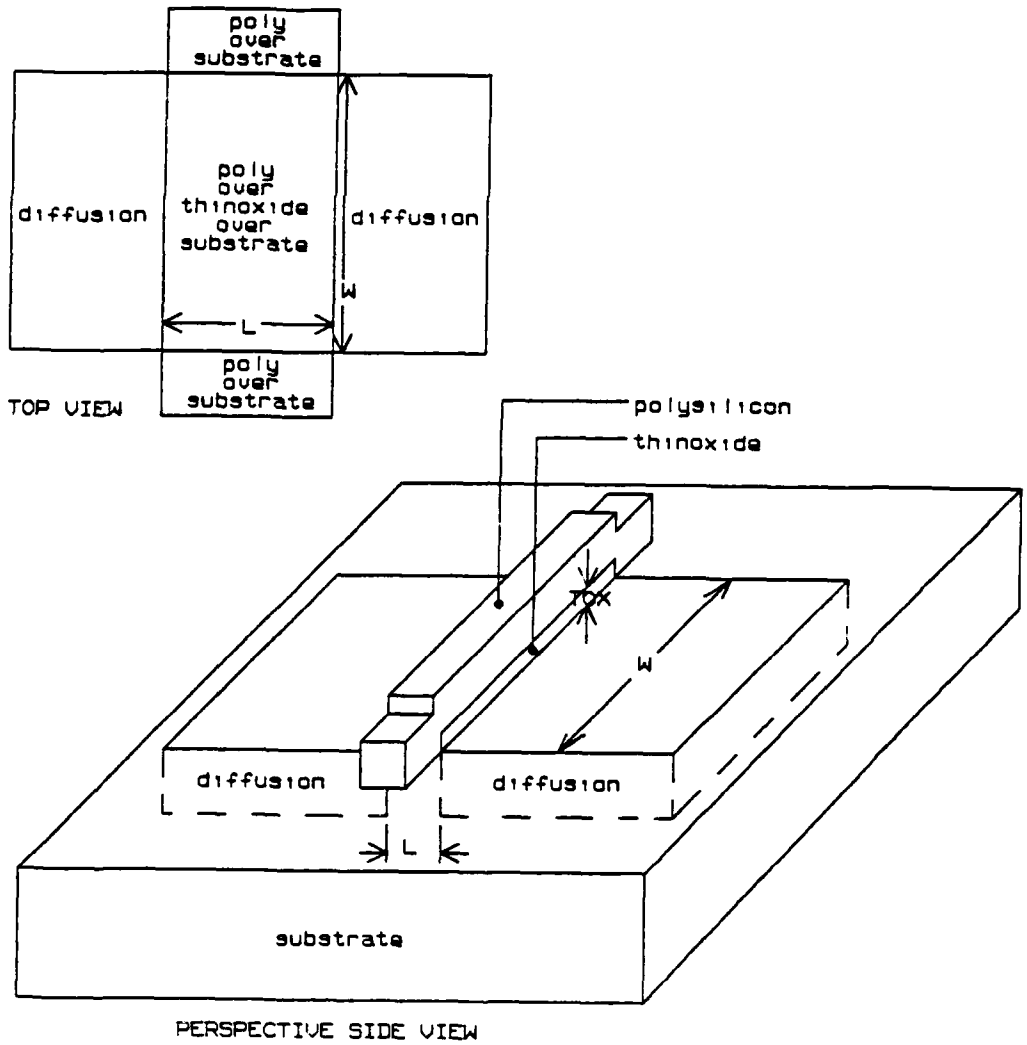


Figure 2.11 Idealized Physical MOSFET

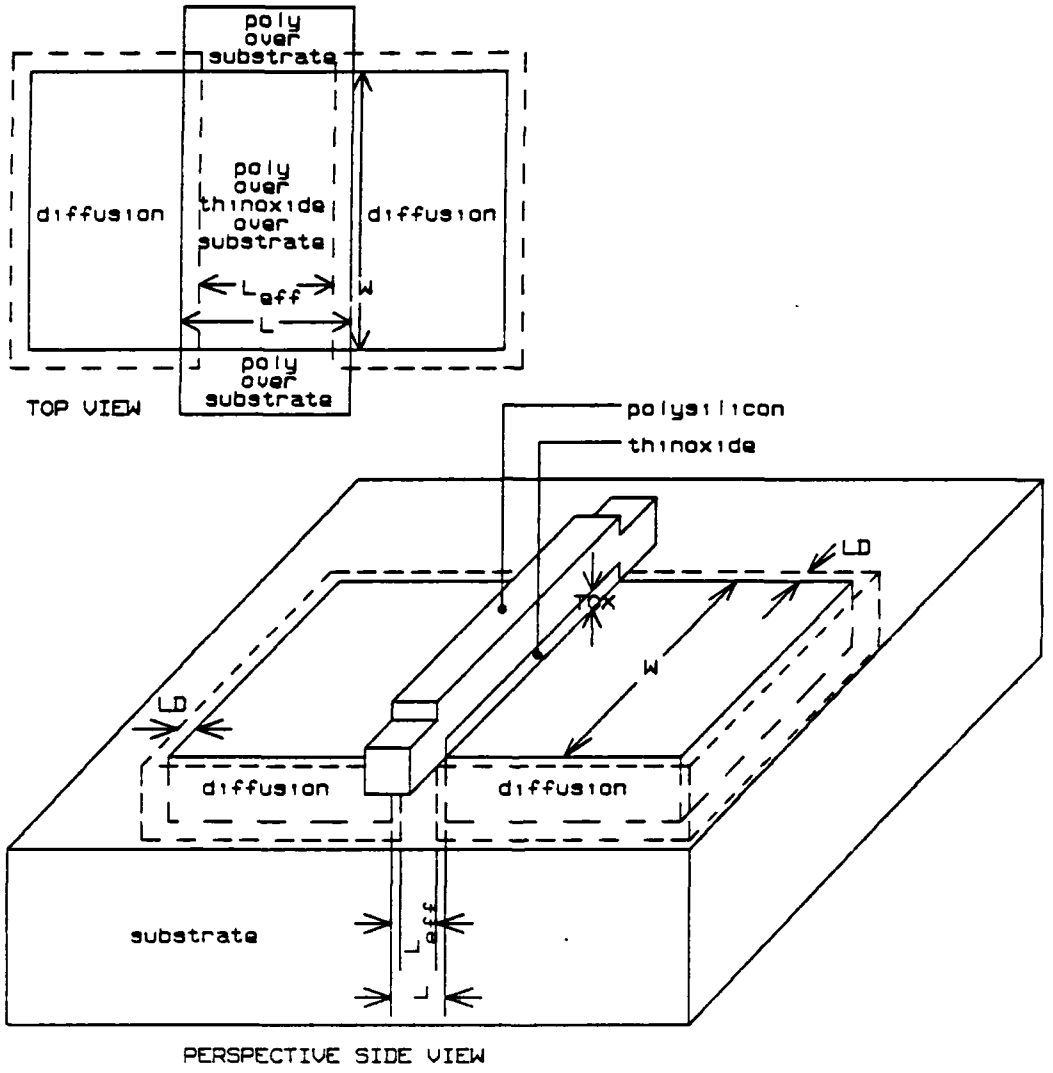


Figure 2.12 Effects of Lateral Diffusion of Doping Ions on a Physical MOSFET

signal to a gate transitions through $Vdd/2$ and the moment that the output affected by that input transitions through $Vdd/2$ [Weste and Eshraghian, 1985, p. 141]:

$$t_{d_{avg}} = \frac{t_r + t_f}{4} \tag{2.16}$$

One of the more attractive feature of CMOS circuits is that symmetric output sourcing and sinking currents are easily achieved resulting in equal rise and fall times. When this is achieved $t_{d_{avg}}$ becomes:

$$t_{d_{avg}} = \frac{t_r}{2} = \frac{t_f}{2} \tag{2.17}$$

To achieve $t_r = t_f$ for the inverter in Figure 2.8 equations (2.14) and (2.15) are set equal to one another then W_n is solved for. The result is:

$$W_n = W_p \cdot A \tag{2.18}$$

where

$$A = \frac{(L_n - 2 \cdot LD_n) \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)}{\left(L_p - 2 \cdot LD_p\right) \cdot (KP_n\{T\}) \cdot (Vdd - Vto_n\{T\})} \cdot \frac{\left[\frac{(Vto_n\{T\} - 0.1 \cdot Vdd)}{\left(Vdd - Vto_n\{T\}\right)} + \frac{1}{2} \cdot \log_e\left(\frac{19 \cdot Vdd - 20 \cdot Vto_n\{T\}}{Vdd}\right)\right]}{\left[\frac{\left(|Vto_p\{T\}| - 0.1 \cdot Vdd\right)}{\left(Vdd - |Vto_p\{T\}|\right)} + \frac{1}{2} \cdot \log_e\left(\frac{19 \cdot Vdd - 20 \cdot |Vto_p\{T\}|}{Vdd}\right)\right]} \tag{2.19}$$

In the drain current equations of Table 2.1 it is evident that as the channel length, L, is decreased drain current increases. To maximize speed, L should be the minimum channel length permitted by the design rules of the particular technology being used. This length is usually the same as the minimum feature size of the technology.

Keeping in mind the previous paragraph on the selection of L_n and L_p, all the values on the right side of equation (2.18) are now known except for W_p. Once W_p is found, a W_n that results in $t_f \simeq t_r$ and $t_{d_{avg}} \simeq \frac{t_f}{2} \simeq \frac{t_r}{2}$ can be readily obtained from equation (2.18). Finding W_p is dealt with in Section 4.

3. Capacitive Loading

The load capacitance, C_L, has been referred to in Section B.2, but not yet described. The term C_L represents three parallel capacitances [Weste and Eshraghian, 1985, p. 123]:

$$C_L = Cbd_{\text{total}} + C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate} + C_r \tag{2.20}$$

where

$Cbd_{\text{total}} \equiv$ the diffusion capacitance of the drain regions connected to the output

$C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate} \equiv$ gate capacitance of the load devices connected to the output of the gate under design

$C_r \equiv$ routing capacitance of the interconnect between the p block, the n block, and the load

It should be noted that in the definition of the term $C_{bd_{\text{total}}}$ the word region is plural. That is to say that in Figure 2.8 each transistor, when active, must drive not only its own drain diffusion to substrate capacitance but also that of the other transistor that is connected to it. This is so because the drains are connected together resulting in parallel capacitances. Cbd_{total}, $C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate}$, and C_r are described in the next three subsections.

a. Diffusion to Substrate Capacitance of the Interconnection Node

Figure 2.13 shows a drain area diffused into the substrate and bloated by an amount LD in all horizontal directions. To the top of the figure would be the channel, but the channel, gate, and source have been removed to make clear that only the drain area is being considered here. At the bottom end of the drain is an area of diffusion that protrudes. This area has been reserved for a metal to diffusion surface contact. Similar contacts may be placed alongside this contact at the bottom end of the drain or on the sides of the drain or both (but obviously not at the channel end). The important drain area dimensions shown in Figure 2.13 are:

$W_{d,eff} \equiv$ Effective diffusion width $= W + 2 \cdot LD$

$L_{d,eff} \equiv$ Effective diffusion length $= L_{d,\mathbf{mask}} + 2 \cdot LD$

where $L_{d,\mathbf{mask}}$ is the mask level diffusion length

$W_{c,eff} \equiv$ effective contact width $= W_c + 2 \cdot LD$

where W_c is the mask level contact width

$L_c \equiv$ mask level contact length

In Figure 2.13 the length of the contact area at the bottom edge of the diffusion runs from top to bottom, and the contact width runs from left to right. If contacts were placed along the sides of the diffusion area shown in Figure 2.13 their lengths would run from left to right in the figure while their widths would run from top to bottom. These dimensions are just the reverse of the dimensions of the contact along the bottom edge of the diffusion area in Figure 2.13.

From Figure 2.13 the drain area is defined:

$$AD = W_{d,eff} \cdot L_{d,eff} + N \cdot W_{c,eff} \cdot L_c \tag{2.21}$$

where N is the number of metal to diffusion contacts placed around the diffusion area.

From Figure 2.13 the drain perimeter is defined:

$$PD = 2 \cdot \left(W_{d,eff} + L_{d,eff} + N \cdot L_c \right) = 2 \cdot W_{d,eff} + 2 \cdot \left(L_{d,eff} + N \cdot L_c \right) \quad (2.22)$$

where N is as defined in equation (2.21)

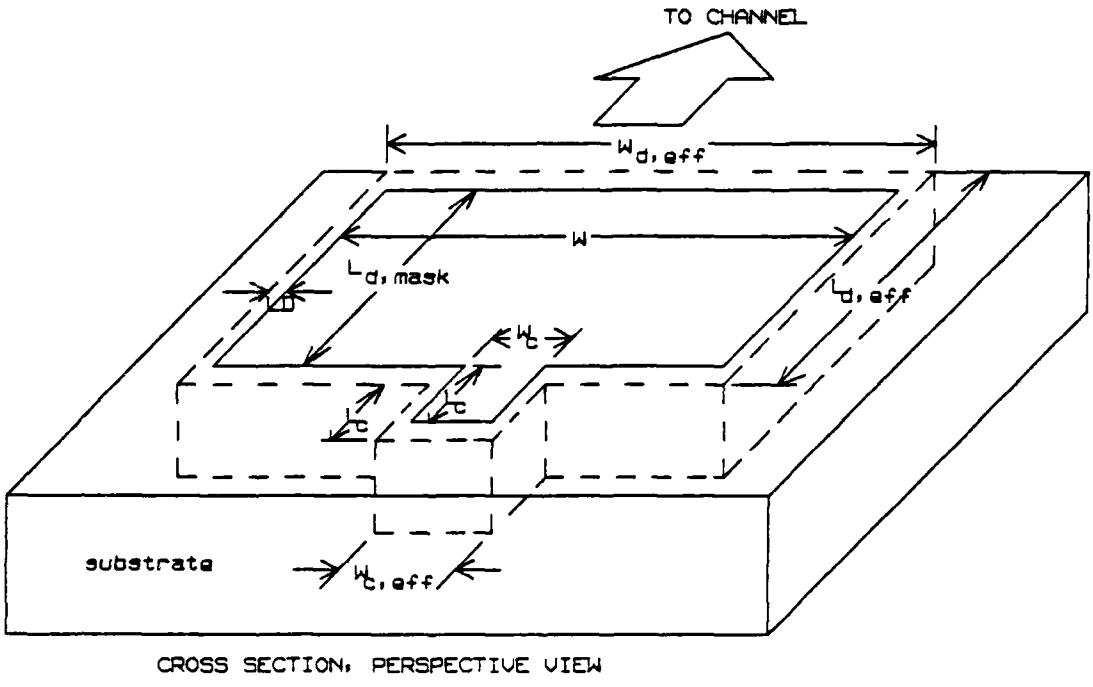


Figure 2.13 Drain Diffusion Area Dimensions

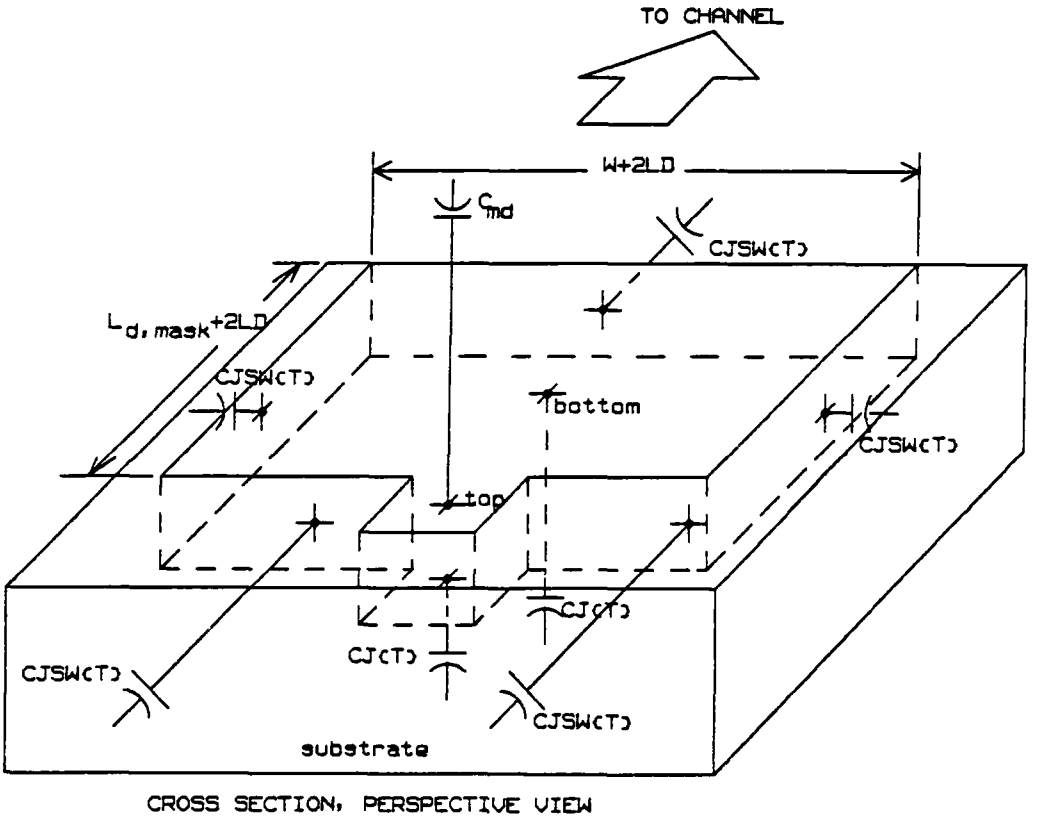


Figure 2.14 Bulk-Drain Depletion Capacitance and Metal to Diffusion Contact Capacitance

Figure 2.14 shows the three capacitances of concern in calculating *Cbd*. $CJ\{T\}$ and $CJSW\{T\}$ in Figure 2.14 are defined in Tables 2.2 and 2.3. C_{md} is the capacitance that results by bringing metal into contact with the protrusion areas—like the one at the bottom edge of the diffusion in Figure 2.14—that are reserved for metal to diffusion contacts.

Having defined the dimensions of the drain diffusion and its capacitive components, Cbd may now be defined [MicroSim Corp., 1987, p. 79]:

$$\begin{aligned} Cbd\{T, Vbd\} = &\quad \text{bulk-drain depletion capacitance} \\ &+ \text{metal to diffusion contact capacitance} \\ = &\quad \text{bottom junction capacitance + sidewall capacitance} \\ &+ \text{metal to diffusion contact capacitance} \\ = &\quad AD \cdot CJ\{T\} \cdot Cbdj\{T, Vbd\} \\ &\quad + PD \cdot CJSW\{T\} \cdot Cbds\{T, Vbd\} \\ &\quad + N \cdot L_c \cdot W_c \cdot C_{md} \end{aligned} \tag{2.23}$$

Cbd is a function of Vbd because $Cbdj$ and $Cbds$ in equation (2.23) are functions of Vbd. This implies that all three will vary during switching due to variations in bias level. To reduce the difficulty that this introduces in the analysis, values that are dependent on temperature only may be assigned to $Cbdj$ and $Cbds$ if the values assigned are chosen carefully and with good judgment. [McCarthy, 1982, p.58]

MicroSim's PSpice defines $Cbdj\{T, Vbd\}$ and $Cbds\{T, Vbd\}$ [MicroSim Corp., 1987, p. 80]:

$$\begin{aligned} Cbdj\{T, Vbd\} &= (1 - FC)^{-(1+MJ)} \\ &\cdot \Big(1 - FC \cdot (1 + MJ) + MJ \cdot |Vbd| / PB\{T\}\Big) [\text{dimensionless}] \end{aligned} \tag{2.24}$$

$$\begin{aligned} Cbds\{T, Vbd\} &= (1 - FC)^{-(1+MJSW)} \\ &\cdot \Big(1 - FC \cdot (1 + MJSW) + MJSW \cdot |Vbd| / PB\{T\}\Big) [\text{dimensionless}] \end{aligned} \tag{2.25}$$

During switching $0.1 \cdot Vdd < |Vbd| < 0.9 \cdot Vdd$. $|Vbd|$ does not dwell at either end of this range any longer than it does at the other. Therefore, $Cbdj$ and $Cbds$ may be expressed as function of T, independent of Vbd by simple averaging:

$$Cbdj_{d_{\mathbf{avg}}}\{T\} = \frac{1}{2} \cdot \left[Cbdj\{T, Vbd\} \Big|_{Vbd=0.9Vdd} + Cbdj\{T, Vbd\} \Big|_{Vbd=0.1 \cdot Vdd} \right] \tag{2.26}$$

$$Cbds_{d_{\mathbf{avg}}}\{T\} = \frac{1}{2} \cdot \left[Cbds\{T, Vbd\} \Big|_{Vbd=0.9Vdd} + Cbdj\{T, Vbd\} \Big|_{Vbd=0.1 \cdot Vdd} \right] \tag{2.27}$$

Substituting equations (2.26) and (2.27) into equation (2.23) allows Cbd to be expressed as a function of temperature only:

$$\begin{aligned} Cbd\{T\} =& AD \cdot CJ\{T\} \cdot Cbdj_{\mathbf{avg}}\{T\} + PD \cdot CJSW\{T\} \cdot Cbds_{\mathbf{avg}}\{T\} \\ &+ N \cdot L_c \cdot W_c \cdot C_{md} \end{aligned} \tag{2.28}$$

Substituting equations (2.21) and (2.22) for the drain area and drain perimeter into equation (2.28) and grouping terms gives:

$$\begin{aligned} Cbd\{T\} =& W_{d,eff} \cdot \left(L_{d,eff} \cdot CJ\{T\} \cdot Cbdj_{\mathbf{avg}}\{T\} + 2 \cdot CJSW\{T\} \cdot Cbds_{\mathbf{avg}}\{T\} \right) \\ &+ 2 \cdot \left(L_{d,eff} + N \cdot L_c \right) \cdot CJSW\{T\} \cdot Cbds_{\mathbf{avg}}\{T\} \\ &+ N \cdot L_c \cdot \left(W_{c,eff} \cdot CJ\{T\} \cdot Cbdj_{\mathbf{avg}}\{T\} + W_c \cdot C_{md} \right) \end{aligned} \tag{2.29}$$

Figure 2.15 is a literal translation from the schematic in Figure 2.8 (with no load attached) to a mask level layout. That is to say that the p and n MOSFETs are in the same position with respect to one another in Figure 2.15 as they are in Figure 2.8. Figure 2.15 shows graphically why each transistor drives

not only the load and its own drain diffusion capacitance but the drain diffusion capacitance of the other transistor as well. It is because the two drain areas are hard-wired together.

Since the two capacitances $Cbd_p\{T\}$ and $Cbd_n\{T\}$ are in parallel with one another $Cbd_{\text{total}}\{T\}$ becomes:

$$Cbd_{\text{total}}\{T\} = Cbd_p\{T\} + Cbd_n\{T\} \tag{2.30}$$

Equation (2.30) can now be fully expanded according to the following procedure:

1.) Expand $W_{d,eff}$, $L_{d,eff}$, and $W_{c,eff}$ in equation (2.29) according to their definitions at the beginning of Section B.3.a.

2.) Take the expression resulting from step 1 and substitute it into equation (2.30).

3.) Make use of equation (2.18) to express n channel widths in terms of p channel widths.

4.) Collect terms.

The result is:

$$Cbd_{\text{total}}\{T\} = Wp \cdot C + D \qquad \text{[farad]} \tag{2.31}$$

where

$$\begin{aligned} C = \Big(& \left[L_{d,\text{mask}_p} + 2 \cdot LD_p\right] \cdot CJ_p\{T\} \cdot Cbdj_{\text{avg},p}\{T\} \\ & + 2 \cdot CJSW_p\{T\} \cdot Cbds_{\text{avg},p}\{T\} \\ & + A \cdot \left[L_{d,\text{mask}_n} + 2 \cdot LD_n\right] \cdot CJ_n\{T\} \cdot Cbdj_{\text{avg},n}\{T\} \\ & + 2 \cdot A \cdot CJSW_n\{T\} \cdot Cbds_{\text{avg},n}\{T\} \Big) \end{aligned} \tag{2.32}$$

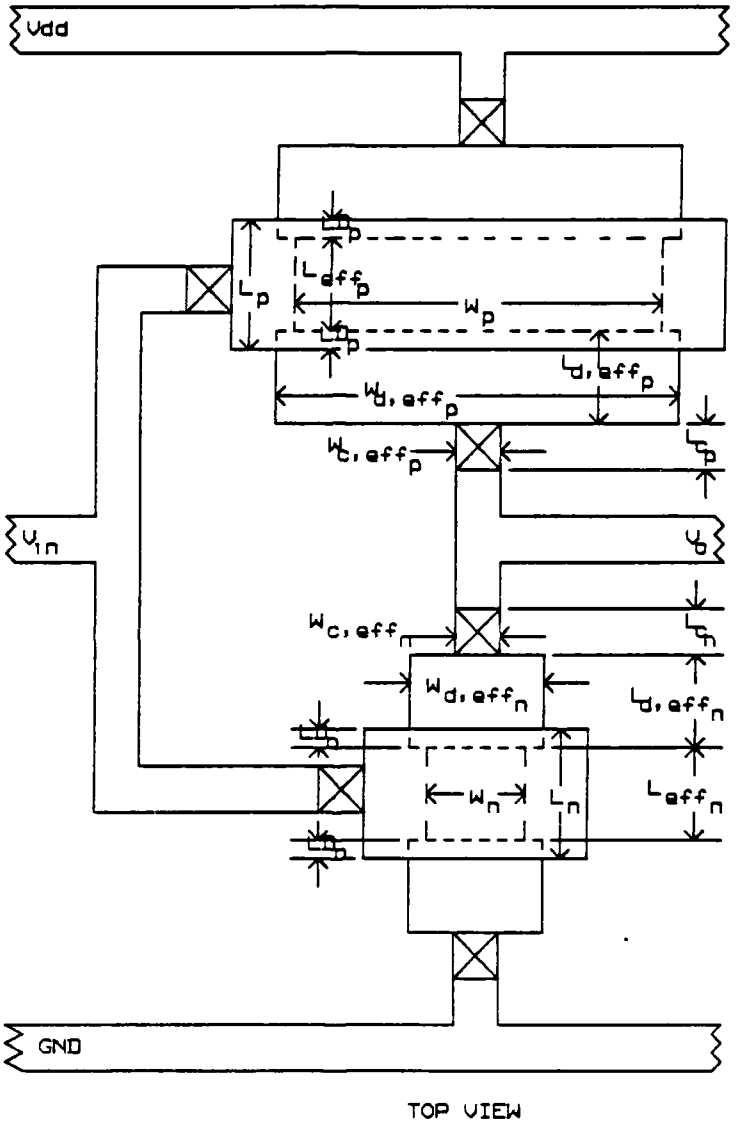


Figure 2.15 Mask Level Layout of a Static CMOS Inverter

$$D = \sum_{x} \Big[2 \cdot LD_x \cdot \Big(\big[L_{d,\,\mathrm{mask}_x} + 2 \cdot LD_x \big] \cdot CJ_x\{T\} \cdot Cbdj_{\mathrm{avg},x}\{T\} + 2 \cdot CJSW_x\{T\} \cdot Cbds_{\mathrm{avg},x}\{T\} \Big) + 2 \cdot \Big(\big[L_{d,\mathrm{mask}_x} + 2 \cdot LD_x \big] + N_x \cdot L_{c_x} \Big) \cdot CSJW_x\{T\} \cdot Cbds_{\mathrm{avg},x}\{T\} + N_x \cdot L_{c_x} \cdot \Big(\big[W_{c_x} + 2 \cdot LD_x \big] \cdot CJ_x\{T\} \cdot Cbdj_{\mathrm{avg},x}\{T\} + W_{c_x} \cdot C_{md_x} \Big) \Big]$$

for $x = p, n$

(2.33)

and A in equation (2.32) is defined by equation (2.19).

b. Gate Capacitance of the Next Stage

The capacitance of the input gates of the immediate next logic stage is determined in this section. Figure 2.16 shows a single load MOSFET from a top view. The load channel dimensions W_{load} and $L_{ch,\mathrm{load,eff}}$ are defined in Figure 2.16 where:

$$L_{ch,\mathrm{load,eff}} = L_{ch,\mathrm{load,mask}} - 2 \cdot LD$$

Figure 2.17 shows three capacitances formed between the gate material above the surface plane of the wafer and the channel and diffusion areas below. The capacitance Cg is the capacitance that forms between the gate and the substrate. This capacitance is much larger when the load transistor is off than when it is on. In design work it is always safest to estimate capacitive loading conservatively so that a realistic safety margin is allowed for variable operating conditions of a physical device. Since the load capacitance component is largest when the load transistor is off it is best from an engineering standpoint to assume that the physical device

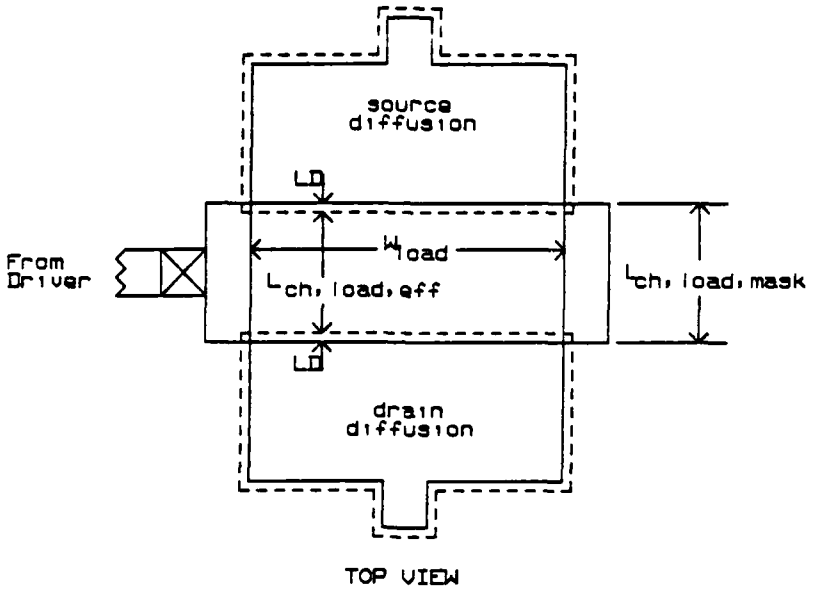


Figure 2.16 Load MOSFET Channel Dimensions

will be operated at low frequency (below 100 Hz).[3] For low frequency operation Cg may be estimated from [Weste and Eshraghian, 1985, p. 125]:

$$Cg = \left(\frac{\epsilon_0 \cdot \epsilon_{\text{rel,oxide}}\{T\}}{TOX}\right) \cdot \left(\left[L_{ch,\text{load, mask}} - 2 \cdot LD\right] \cdot W_{\text{load}}\right) \qquad (2.34)$$

Cgs and Cgd in Figure 2.17 are the capacitances between the gate and the source diffusion and between the gate and drain diffusion. These capacitances are a direct result of the overlapping of the gate and drain and of the gate and source caused by the lateral diffusion of doping ions into the intended channel area

[3] Note that channel length becomes shorter when the load device operates in the saturation region. In saturation, $L_{ch,\text{load},eff} = (L_{ch,\text{load,mask}} - 2 \cdot LD) \cdot (1 - \text{LAMBDA} \cdot |V_{ds}|)$. Since the load device operates in all three regions—cutoff, linear, and saturation—the region chosen for modeling the physical load channel length is that which yields the most conservative estimate for Cg, namely the linear region where $L_{ch,\text{load},eff} = L_{ch,\text{load,mask}} - 2 \cdot LD$.

as discussed in Section B.2. Cgs and Cgd are determined from [MicroSim Corp., 1987, p. 80]:

$$Cgs = CGSO \cdot W_{\text{load}}$$

$$Cgd = CGDO \cdot W_{\text{load}}$$

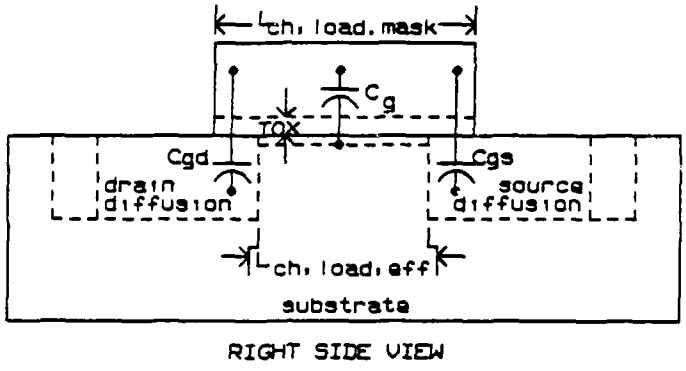


Figure 2.17 Load MOSFET Gate Capacitance

As Cg, Cgs, and Cgd are in parallel with one another, $C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate}$ is equal to their sum over all load devices:

$$C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate} = \sum_{\substack{\text{load}\\\text{devices}}} [Cg + Cgs + Cgd] \tag{2.34}$$

$$= \sum_{\substack{\text{load}\\\text{devices}}} \left[\left(\frac{\epsilon_0 \cdot \epsilon_{\text{rel,oxide}}\{T\}}{TOX} \right) \cdot \left(L_{ch,\text{load,mask}} - 2 \cdot LD \right) \cdot W_{\text{load}} + CGSO \cdot W_{\text{load}} + CGDO \cdot W_{\text{load}} \right] \tag{2.35}$$

where all the variables in equation (2.35) are defined in Tables 2.2 and 2.3.

c. Routing Capacitance

Routing capacitance between layers or between a layer and substrate is determined by multiplying the capacitance per unit area measured between the specific layers by the area that is jointly occupied by both layers. Values for these unit area capacitances must be obtained from the fabrication facility being used. Typical values for a 4 micron, two level metal, silicon gate CMOS process are provided in Weste and Eshraghian [Weste and Eshraghian, 1985, p. 135]. An alternate method is to remove C_r from equation (2.20) and compute the delay due to routing capacitance separately. A paper by John L. Wyatt, Jr. entitled, "The Practical Engineer's No-Nonsense Guide to On-Chip Signal Delay Calculation," available through the M.I.T. VLSI memo series gives worst case bounds on delay time due to routing for use in this type of approach [Wyatt, 1987]. Note that the capacitance due to diffusion to metal contacts on the drains wired to the node that connects the p and n blocks is considered separate from C_r, and is developed in Section B.3.a.

4. Finding W_p and W_n

Figure 2.15, which was discussed in Section B.3.a, shows the dimensions W_p and W_n. As discussed in Section B.1, the circuit designer needs to have a method to determine the widths W_p and W_n required to drive the parasitics and known capacitive load within a designer specified time constraint given some expected temperature and supply voltage operating conditions. All that has been accomplished thus far in Chapter II is to gather together the tools necessary to make this calculation. In this section the tools are used together to find W_p and W_n for the static CMOS inverter of Figures 2.8 and 2.15.

Rearranging equation (2.15) to solve for W_p gives:

$$W_p = F \cdot C_L \quad [G + H] \tag{2.36}$$

where:

$$F = \frac{2 \cdot (L_p - 2 \cdot LD_p)}{(t_r) \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)} \tag{2.37}$$

$$G = \frac{(|Vto_p\{T\}| - 0.1 \cdot Vdd)}{(Vdd - |Vto_p\{T\}|)} \tag{2.38}$$

$$H = \frac{1}{2} \cdot \log_e \left(\frac{19 \cdot Vdd - 20 \cdot |Vto_p\{T\}|}{Vdd} \right) \tag{2.39}$$

But C_L is itself a function of the unknown W_p since

$$C_L = Cbd_{\text{total}} + C \sum_{\substack{\text{load} \\ \text{devices}}} \text{gate} + C_r \qquad \text{; from equation (2.20)} \tag{2.40}$$

$$C_L = C \cdot W_p + D + C \sum_{\substack{\text{load} \\ \text{devices}}} \text{gate} + C_r \qquad \text{; from equation (2.31)} \tag{2.41}$$

Renaming $C \sum_{\substack{\text{load} \\ \text{devices}}} \text{gate}$ and C_r for simplicity:

$$B = C \sum_{\substack{\text{load} \\ \text{devices}}} \text{gate} \tag{2.42}$$

$$E = C_r \tag{2.43}$$

Substituting equations (2.42) and (2.43) into equation (2.41) gives:

$$C_L = C \cdot W_p + D + B + E \tag{2.44}$$

Substituting equation (2.44) into equation (2.36) and solving for W_p gives the final result for the static CMOS inverter:

$$W_p = \frac{F \cdot (G + H) \cdot (B + D + E)}{1 - F \cdot C \cdot (G + H)} \tag{2.45}$$

$$W_n = W_p \cdot A \text{ (from equation (2.18))} \tag{2.46}$$

The equations defining the variables in equations (2.45) and (2.46) are referenced in Table 2.4.

Practical calculation of W_p requires a computer program. A program to calculate W_p, W_n, and all other MOSFET parameters required on a MOSFET card in a PSpice deck is provided in Appendix A. This program is written for an HP41CX with card reader and two extended memory modules.

TABLE 2.4 IMPORTANT EQUATIONS FOR THE STATIC CMOS INVERTER

Variable	Defining Equations(s)
W_p	(2.45)
W_n	(2.46)
A	(2.19)
B	(2.42) and (2.35)
C	(2.32)
D	(2.33)
E	(2.43)
F	(2.37)
G	(2.38)
H	(2.39)

5. Drain Resistance

With respect to the drain diffusion region, only its capacitive value has been determined thus far. Proper modeling of a MOSFET in PSpice requires that the drain resistance be provided as well. Figure 2.18 shows a simple drain diffusion region (substrate omitted from the figure) and its resistive equivalent, RD. The resistance between end faces of the figure can be calculated from [McCarthy, 1982, p. 39]:

$$RD = \overline{\rho} \cdot \frac{\text{drain length}}{\text{end face area}} \tag{2.47}$$

where $\overline{\rho}$ is the average resistivity for the diffusion layer.

In terms of SPICE parameters [McCarthy, 1982, p. 59]:

$$RSH = \frac{\overline{\rho}}{XJ} \tag{2.48}$$

Therefore,

$$RD = \overline{\rho} \cdot \left(\frac{L_{d,eff}}{W_{d,eff} \cdot XJ} \right)$$

$$RD = RSH \cdot XJ \cdot \left(\frac{L_{d,eff}}{W_{d,eff} \cdot XJ} \right)$$

$$RD = RSH \cdot \left(\frac{L_{d,mask} + 2 \cdot LD}{W + 2 \cdot LD} \right) \tag{2.49}$$

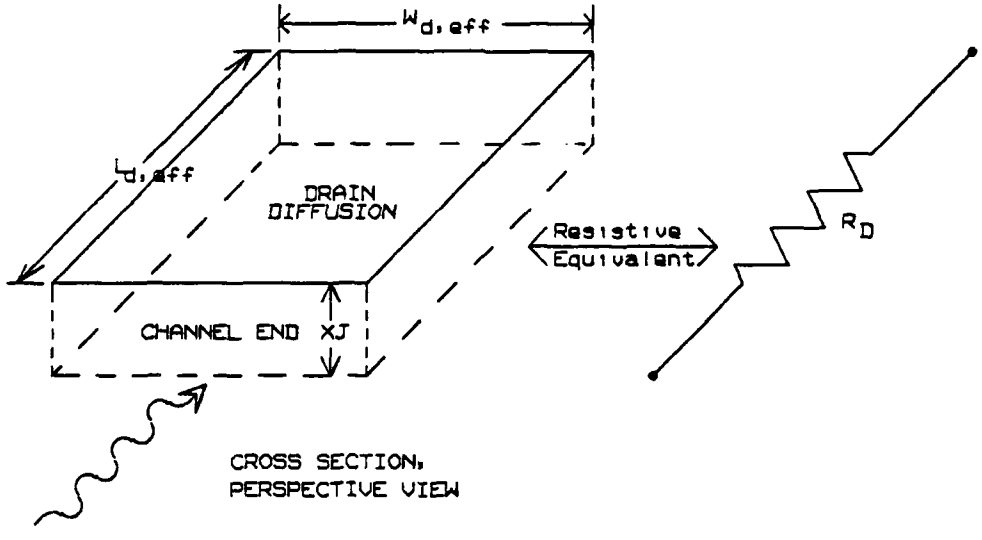


Figure 2.18 Drain Diffusion Region and Resistive Equivalent

Figure 2.19 shows a drain diffusion area with a single contact area defined and a resistive equivalent. Since R and R_c in Figure 2.19 are in series:

$$RD = R + R_c \tag{2.50}$$

$$RD = \overline{\rho} \cdot \left(\frac{L_{d,eff}}{W_{d,eff} \cdot XJ} \right) + \overline{\rho} \cdot \left(\frac{L_c}{W_{c,eff} \cdot XJ} \right)$$

$$RD = RSH \cdot \left[\frac{(L_{d,mask} + 2 \cdot LD)}{(W + 2 \cdot LD)} + \frac{L_c}{W_c + 2 \cdot LD} \right] \tag{2.51}$$

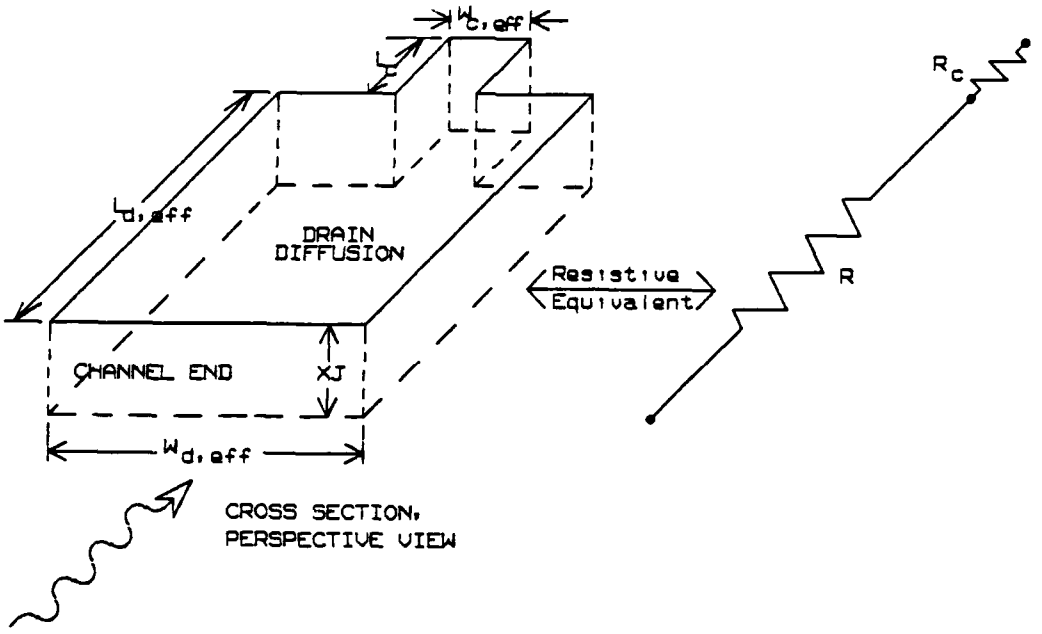


Figure 2.19 Drain Diffusion Region with a Single Metal to Diffusion Contact Area and Resistive Equivalent

Figure 2.20 is an extension of the resistance calculations performed in equations (2.49) and (2.51) to the general case where several contact areas may be placed around the diffusion area. In the figure

$$R_{eq} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \cdots \frac{1}{R_N}}$$

But assuming that all the contact areas have identical dimensions, $R_1 = R_2 = \cdots = R_N = R_c$, where R_c is the resistance of any one contact area and so:

$$R_{eq} = \frac{1}{N} \cdot R_c \tag{2.52}$$

Using the results of equations (2.52) and (2.51) gives the drain resistance for the general case depicted in Figure 2.20:

$$RD = R + R_{eq} = RSH \cdot \left[\left(\frac{L_{d,\text{mask}} + 2 \cdot LD}{W + 2 \cdot LD} \right) + \frac{1}{N} \cdot \left(\frac{L_c}{W_c + 2 \cdot LD} \right) \right] \tag{2.53}$$

Given that the contact areas employed by the designer are of the minimum dimensions permitted by the design rules for the technology being used, equations (2.23) in Section B.3.a. and (2.53) in this section confirm an intuitive trade off:

- placing more contact areas decreases drain resistance but increases drain capacitance.
- placing fewer contact areas decreases drain capacitance but increases drain resistance.

6. Theoretical Bounds on Inverter Performance

Equation (2.45) can be manipulated algebraically into the form:

$$W_p = \frac{\left(F \cdot t_r \right) \cdot (G + H) \cdot (B + D + E)}{t_r - \frac{2 \cdot (L_p - 2 \cdot LD_p) \cdot C \cdot (G + H)}{\left(KP_p\{T\} \right) \cdot \left(Vdd - |Vto_p\{T\}| \right)}} \tag{2.54}$$

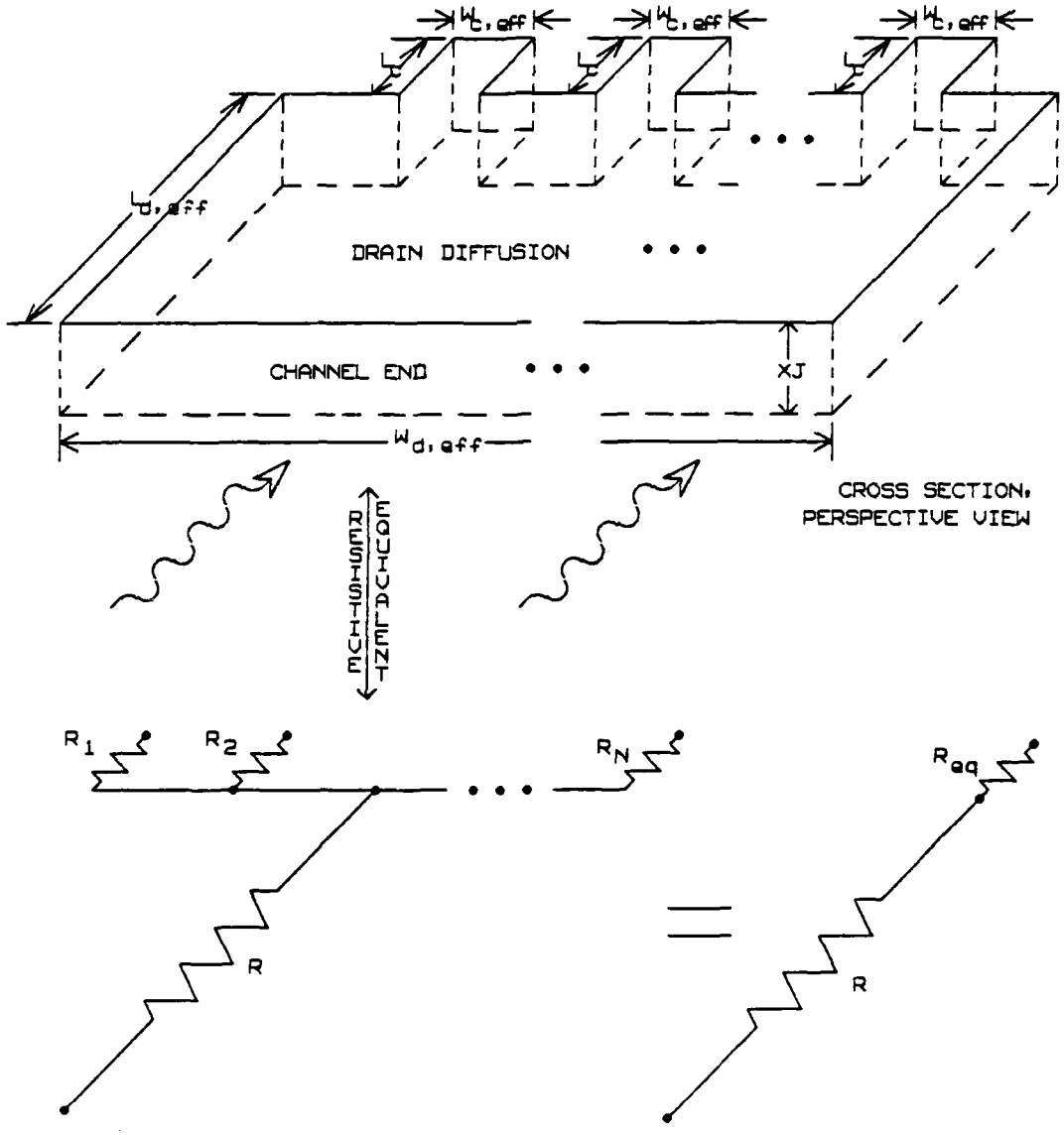


Figure 2.20 Drain Diffusion Region with N Metal to Diffusion Contact Areas and Resistive Equivalent

The denominator of equation (2.54) shows that the p transistor width, W_p, goes to infinity as the denominator goes to zero which occurs at the point where:

$$t_r = \frac{2 \cdot (L_p - 2 \cdot LD_p) \cdot C \cdot (G + H)}{(KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)} \tag{2.55}$$

Obviously the p MOSFET required to source enough current to achieve the rise time defined by equation (2.55) would be so large as to be physically impossible to construct. Therefore, equation (2.55) imposes a theoretical bound on the rise time achievable by a static CMOS inverter of the type shown in Figures 2.8 and 2.15.

If a previously constructed inverter is being examined its minimum theoretical rise time can be determined by rearranging equation (2.45) into the following form.

$$t_r = \frac{2 \cdot (L_p - 2 \cdot LD_p) \cdot (G + H)}{(KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)} \cdot \left[\frac{(B + D + E)}{W_p} + C \right] \tag{2.56}$$

It is important to realize that the inverter being examined may have been constructed such that the variable A in the expression $W_n = W_p \cdot A$ is not defined by equation (2.19). In such a case, the numerical value of A that is obtained from the physical layout of the inverter should be used in the calculation of the variable C in equation (2.56).

For this same inverter that has already been constructed the amount of capacitance that it can drive in t_r seconds can be determined theoretically by manipulating equation (2.45) into the following form:

$$B + E = W_p \cdot \left[\frac{(t_r) \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)}{2 \cdot (L_p - 2 \cdot LD_p) \cdot (G + H)} - C \right] - D \tag{2.57}$$

The same caveat about the variable A that was applied to equation (2.56) is equally applicable to equation (2.57).

C. EXTENSION OF THE STATIC CMOS INVERTER TRANSISTOR SIZING EQUATIONS TO GENERAL STATIC CMOS LOGIC DESIGN

The static CMOS inverter in Figures 2.8 and 2.15 involved only a single p MOSFET in the p block and a single n MOSFET in the n block. Figure 2.21 shows a compound gate that implements the logic function $F = \overline{((A \cdot B) + (C \cdot D))}$ [Weste and Eshraghian, 1985, p. 15]. There are two parallel combinations of p MOSFETs in the p block of this compound gate. The two parallel combinations are in series with one another. A similar situation exists in the n block. To apply the equations developed in Section B to a problem like translating from the schematic in Figure 2.21 to a mask level layout requires that some of the variables in equation (2.45) be modified.

Equation (2.30) is the basis for computing $Cbd_{\text{total}}\{T\}$ in the static CMOS inverter of Section B. Section B.3 states that $Cbd_{\text{total}}\{T\}$ involves all of the drain regions hard wired to the interconnection node from which the output is taken. For the static CMOS inverter of Section B this involves only one drain in the p block and one drain in the n block. For a compound gate $Cbd_{\text{total}}\{T\}$ can involve considerably more drain diffusion area than that. For example, in Figure 2.21 there are four drain regions wired to the interconnection node. If signals A and C are low and signals B and D are high in Figure 2.21, the p MOSFET fed by signal A will have to drive the load capacitance, the capacitance of its own drain, and the capacitance of the other three drains wired to the interconnection node as well. Therefore, equation (2.30) is modified for general static CMOS logic design:

$$Cbd_{\text{total}} = \#_p \cdot Cbd_p\{T\} + \#_n \cdot Cbd_n\{T\} \tag{2.58}$$

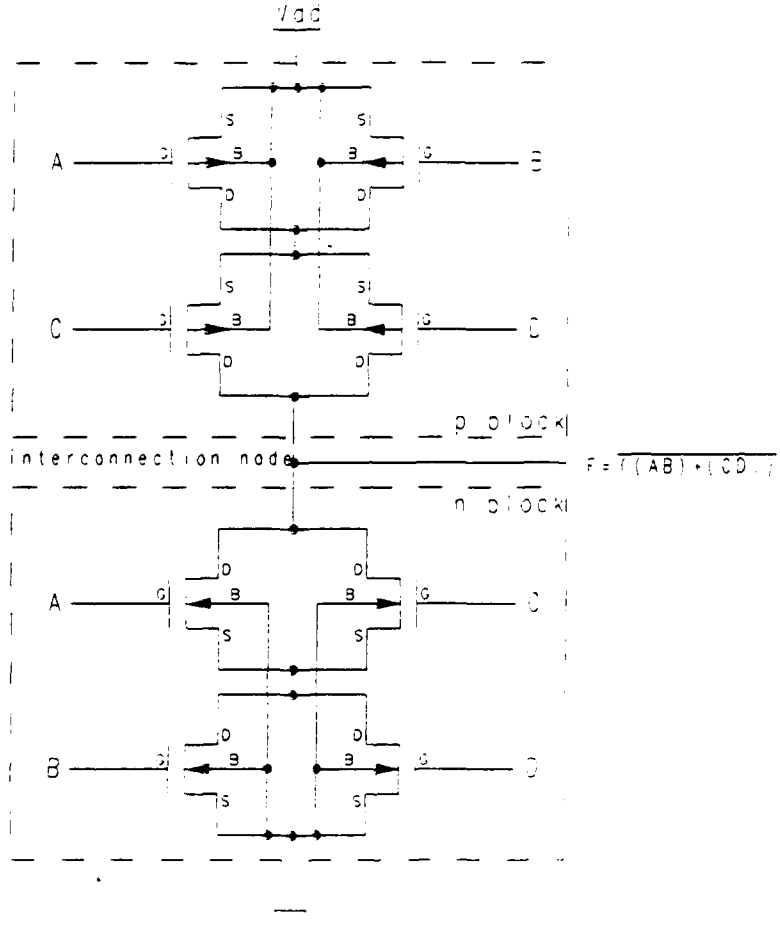


Figure 2.21 Static CMOS Compound Gate Implementation of $F = \overline{((A \cdot B) + (C \cdot D))}$.

where

$\#_p \equiv$ the total number of p drains attached to the interconnection node between the p and n blocks, each with identical dimensions.

$\#_n \equiv$ the total number of n drains attached to the interconnection node between the n and p blocks, each with identical dimensions.

Equation (2.58) can now be fully expanded according to the same procedure used to expand equation (2.30) in Section B.3.a. The result is that:

$$Cbd_{\text{total}}\{T\} = W_p \cdot C' + D' \qquad [farad] \tag{2.59}$$

where C' and D' are no longer defined by equations (2.32) and (2.33):

$$\begin{aligned} C' = \Big(& \#_p \cdot \left[L_{d,\text{mask}_p} + 2 \cdot LD_p\right] \cdot CJ_p\{T\} \cdot Cbdj_{\text{avg},p}\{T\} \\ & + \#_p \cdot 2 \cdot CJSW_p\{T\} \cdot Cbds_{\text{avg},p}\{T\} \\ & + \#_n \cdot A' \cdot \left[L_{d,\text{mask}_n} + 2 \cdot LD_n\right] \cdot CJ_n\{T\} \cdot Cbdj_{\text{avg},n}\{T\} \\ & + \#_n \cdot A' \cdot 2 \cdot CJSW_n\{T\} \cdot Cbds_{\text{avg},n}\{T\} \Big) \end{aligned} \tag{2.60}$$

where A' is defined later in equation (2.64).

$$\begin{aligned} D' = \sum_x \#_x \cdot \Big[& 2 \cdot LD_x \cdot \Big(\left[L_{d,\ \text{mask}_x} + 2 \cdot LD_x\right] \cdot CJ_x\{T\} \cdot Cbdj_{\text{avg},x}\{T\} \\ & + 2 \cdot CJSW_x\{T\} \cdot Cbds_{\text{avg},x}\{T\} \Big) \\ & + 2 \cdot \Big(\left[L_{d,\text{mask}_x} + 2 \cdot LD_x\right] + N_x \cdot L_{c_x} \Big) \cdot CSJW_x\{T\} \cdot Cbds_{\text{avg},x}\{T\} \\ & + N_x \cdot L_{c_x} \cdot \Big(\left[W_{c_x} + 2 \cdot LD_x\right] \cdot CJ_x\{T\} \cdot Cbdj_{\text{avg},x}\{T\} + W_{c_x} \cdot C_{md_x} \Big) \Big] \end{aligned}$$

for $x = p, n$

(2.61)

For each p transistor added in series between the supply voltage and the load the rise time at the load increases. If a single p MOSFET has the ability to drive

a load high in t_r seconds then placing k identical p MOSFETs in series would increase the rise time to $k \cdot t_r$ seconds. Similarly, placing m identical n MOSFETs in series between a load and ground increases fall time from t_f seconds to $m \cdot t_f$ seconds [Weste and Eshraghian, 1985, p. 181]. The reason for this is quite apparent from Table 2.1. Placing transistors in series is analogous to increasing the effective channel length of a single transistor. In Table 2.1 effective channel length is found in the denominator of the drain current equations. Therefore increasing effective channel length decreases drain current which implies longer rise and fall times.

In Section B.4 the variable F was calculated based on equation (2.15). Equation (2.15) was calculated for a static CMOS inverter where only a single p MOSFET stood between the supply voltage and the load. As Figure 2.21 illustrates, there may in fact be a long line of p MOSFETs in series between the supply voltage and the load. Therefore the effective channel length in equation (2.15) should be modified for general static CMOS logic design:

$$L_{eff_p} = \sum_{\substack{\text{longest} \\ \text{path to } Vdd}} (L_p - 2 \cdot LD_p) \tag{2.62}$$

where the summation is over the number of p MOSFETs that comprise the longest path between the supply voltage and the load and L_p is assumed to be the minimum mask level channel length permitted by the technology in use.

The change in equation (2.15) is reflected in the calculation of the variable F' for general static CMOS logic design:

$$F' = \frac{2 \cdot \sum_{\substack{\text{longest} \\ \text{pathto } Vdd}} (L_p - 2 \cdot LD_p)}{(t_r) \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)} \tag{2.63}$$

The change in effective channel length also makes it necessary to modify the variable A found in equation (2.19) for use in general static CMOS logic design.

Applying the same reasoning that was used to develop equation (2.63) to the calculation of the variable A gives:

$$A' = \frac{\left[\sum_{\substack{\text{longest} \\ \text{path to } GND}} (L_n - 2 \cdot LD_n)\right] \cdot (KP_p\{T\}) \cdot (Vdd - |Vto_p\{T\}|)}{\left[\sum_{\substack{\text{longest} \\ \text{path to } Vdd}} (L_p - 2 \cdot LD_p)\right] \cdot (KP_n\{T\}) \cdot (Vdd - Vto_n\{T\})} \cdot \frac{\left[\frac{(Vto_n\{T\} - 0.1 \cdot Vdd)}{(Vdd - Vto_n\{T\})} + \frac{1}{2} \cdot \log_e\left(\frac{19 \cdot Vdd - 20 \cdot Vto_n\{T\}}{Vdd}\right)\right]}{\left[\frac{(|Vto_p\{T\}| - 0.1 \cdot Vdd)}{(Vdd - |Vto_p\{T\}|)} + \frac{1}{2} \cdot \log_e\left(\frac{19 \cdot Vdd - 20 \cdot |Vto_p\{T\}|}{Vdd}\right)\right]} \quad (2.64)$$

Incorporating these modifications into equations (2.45) and (2.46) gives the final result for general static CMOS logic design:

$$W_p = \frac{F' \cdot (G + H) \cdot (B + D' + E)}{1 - F' \cdot C' \cdot (G + H)} \quad (2.65)$$

$$W_n = W_p \cdot A' \quad (2.66)$$

The equations defining the variables in equations (2.65) and (2.66) are referenced in Table 2.5.

The W_p solved for is the mask level width of the channel of each p MOSFET connected directly to the interconnection node between the p and n blocks. This width must be maintained for all MOSFETs between the loadmost p MOSFETs and the supply voltage. Similarly, the W_n solved for is the mask level width of the channel of each n MOSFET connected directly to the interconnection node between the n and p blocks. This width must be maintained for all n MOSFETs between the loadmost n MOSFETs and ground.

TABLE 2.5 IMPORTANT EQUATIONS FOR GENERAL STATIC CMOS LOGIC DESIGN

Variable	Defining Equations(s)
W_p	(2.65)
W_p	(2.66)
A^p	(2.64)
B	(2.42) and (2.35)
C′	(2.60)
D′	(2.61)
E	(2.43)
F′	(2.63)
G	(2.38)
H	(2.39)

D. EXTENSION OF THE TRANSISTOR SIZING EQUATIONS TO NON-RECTANGULAR TRANSISTORS.

To this point the equations developed for transistor sizing have been discussed only in the context of rectangular shaped transistors of the sort depicted in Figure 2.11. It is possible to apply the result obtained to other configurations such as the Manhattan geometry star in Figure 2.22 and the circular transistor in Figure 2.23. The equations in Sections B and C are based on rectangular shaped transistors but only in the sense that total channel lengths and widths and total diffusion lengths, widths, perimeters and areas were defined for rectangular areas. As long as the values assigned to these model parameters are maintained in construction it is possible to apply the results for the rectangular configuration to any configuration. For example, in Figure 2.23 the mask level channel length is $(r_2 - r_1)$ and the mask level channel width is $r_2 \cdot (2 \cdot \pi - \theta)$. These values should be made to equal L_p and W_p from the rectangular model. Similarly, the drain area should be made to equal that of the rectangular configuration. In this way the results of Sections B and C become powerful design tools.

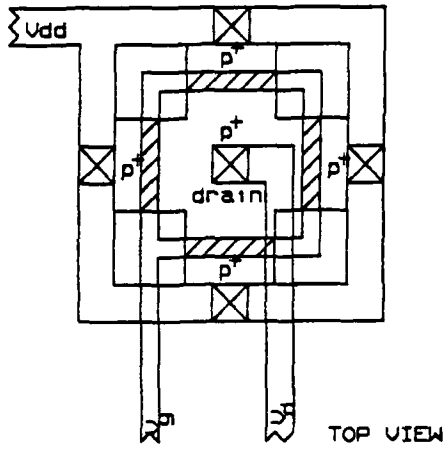


Figure 2.22 Manhattan Geometry Star Configured MOSFET

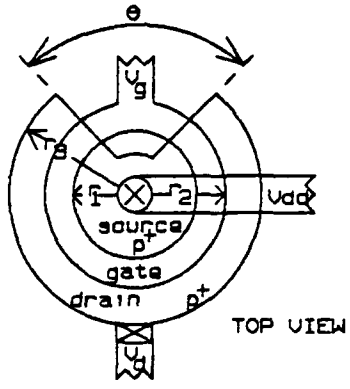


Figure 2.23 Circular MOSFET

E. CHAPTER SUMMARY

The equations presented in this chapter enable one to size transistors to meet desired timing performance standards under known loading conditions, operating temperature, and supply voltage. The equations developed for the static CMOS inverter are implemented in a computer program that is included in Appendix A.

This program is used to apply the equations to a design problem, the design of the super buffer, in the next chapter.

III. DESIGN OF A HIGH-SPEED STATIC CMOS SUPER BUFFER FOR THE MONTEREY SILICON COMPILER

A. BACKGROUND.

A need exists to create and insert into the Monterey Silicon Compiler code a mask level integrated circuit design of a static CMOS super buffer [Malagon, 1987, p. 95]. To understand the role of a super buffer it is necessary to first discuss the circuit that it is a part of. From Sedra and Smith:

> "A critical aspect of the design of any logic device, particularly those using LSI (Large-Scale Integration) and VLSI (Very-Large-Scale Integration) circuit technologies, is the provision of clock signals. Difficulties arise because the clock signal (whether generated on or off the chip) must usually feed many parts of the circuit. This need for large fan-out can be met by applying the clock signal to a string of cascaded inverters and feeding the output of each inverter to a different part of the circuit, a structure known as a *branching fan-out tree.* However, problems may arise with this approach as a result of the differing time delays that the clock signal experiences on the paths to the various parts of the system. To be be specific, if two physically remote segments of a logic network must intercommunicate, it is essential that the relative time variation, or *skew*, of their clocks be controlled and limited to ensure reliable operation.
>
> The problem in VLSI circuit design is further compounded because large clock fan-out and long clock lines imply large capacitive loads and two conflicting dangers: very slow rise and fall times if drive current is inadequate or, alternatively, enormous charging and discharging currents if the driver is too capable and fast charging. Thus, the design of clock systems in VLSI circuits is a challenging problem, involving control of signals that are both large in amplitude and in rate of rise." [Sedra and Smith, 1982, p. 759]

In the Monterey Silicon Compiler architecture the effects of skew are minimized through the use of a two phase non-overlapping clocking scheme [Mullarky, 1987, pp. 11–23]. The need for large fan-out in the clock circuit is satisfied by the super buffer. The super buffer is that part of the branching fan-out tree that feeds clock signals from the pad where off-chip generated clock signals arrive, to the

data storage registers on the chip. Since the super buffer is a gate in the path of incoming clock signals, it must be guaranteed to function within specified timing constraints under given loading lest the operation of an entire chip fail due to the inability of the clock circuit to drive the on chip storage registers within an acceptable amount of time delay. Since a super buffer's sole function is to increase the drive of a propagating signal it can be constructed by cascading two inverters.

Professor Richard W. Hamming of the Naval Postgraduate School said about systems engineering, "If you optimize one component of a system you will hate yourself in the morning." The idea behind the statement is that each component of a system must be designed to fit in well with all the other system components. The clock input pads, super buffers, and storage registers form a complete subsystem within a Monterey Silicon Compiler generated circuit. CMOS replacements for the NMOS clock input pads and the NMOS storage registers already exist, therefore, the replacement CMOS super buffer must be designed with this previous work in mind.

Figure 3.1 demonstrates the concept of employment of super buffers within the Monterey Silicon Compiler architecture. In the figure each super buffer can be seen to consist of two cascaded inverters. Since the clocking scheme used involves two clocks, two clock-signals must be delivered to each register. Thus, the super buffers must be employed in pairs as can be seen in Figure 3.1. Each pair supplies both clock signals to a bank of data registers that straddles the data path.

In Figure 3.1 the inverter symbols within each super buffer are drawn so that the output stage inverter is larger than the input stage inverter. This is intended to be a reflection of the physical construction necessary to achieve the desired electrical behavior of the super buffer. The super buffer output stage must be capable of driving a large fan-out. The input stage must be capable of driving

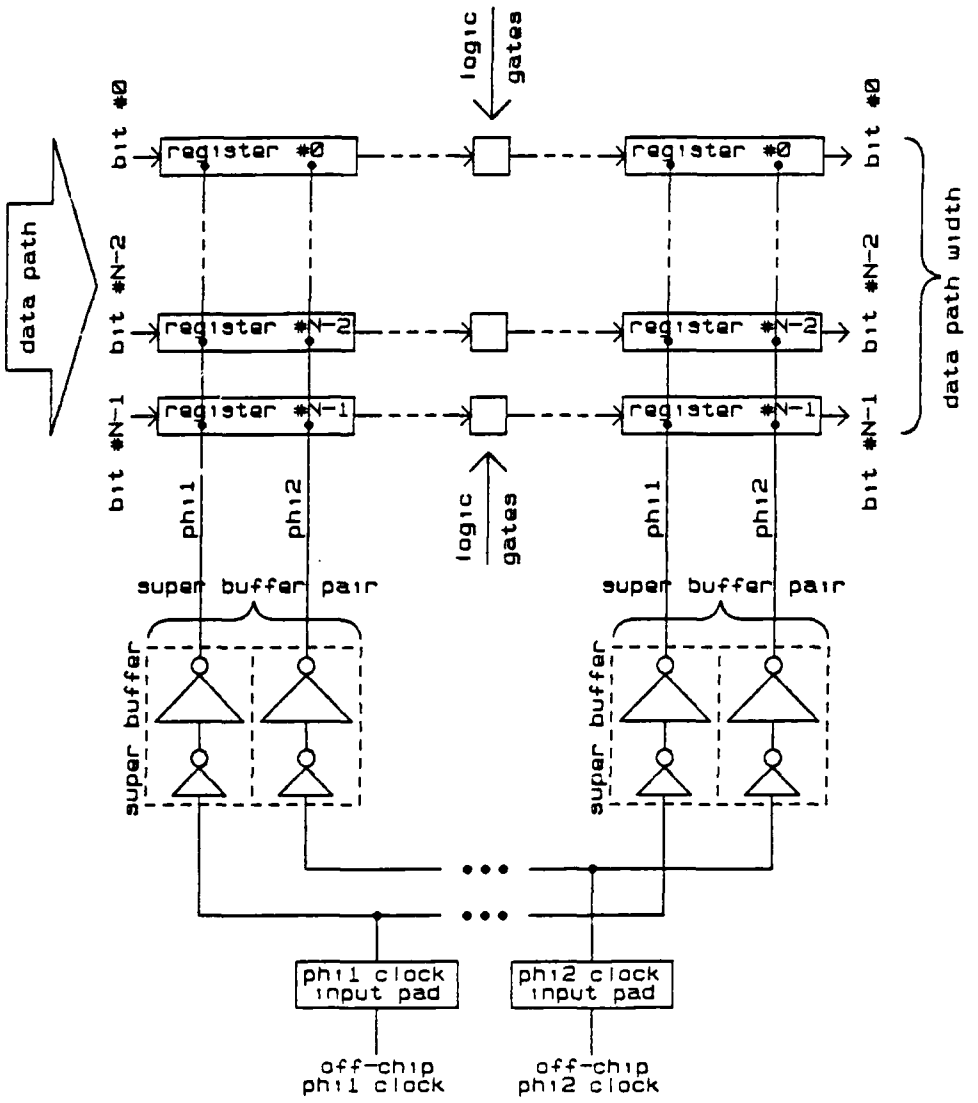


Figure 3.1 Employment of Super Buffer within the Monterey Silicon Compiler Architecture

the gate capacitance of the output stage and at the same time present an input capacitance smaller than the output stage input capacitance. The MOSFETs in the output stage must, therefore, be larger than those in the input stage.

The design task is to find all of the dimensions of the input and output stage inverters given the following:

- worst-case circuit operating temperature
- worst-case supply voltage
- worst-case gate voltage
- worst-case SPICE fabrication parameters

The equations developed in Chapter II are ideally suited to solving this problem. The procedure is documented in the next section. Worst-case conditions are referred to above because worst-case design is essential for reliable circuit operation. The performance of the super buffer with respect to time is of the utmost concern in this design. Therefore, the worst-case parameters to be used in the design process are those that restrict current flow and thereby slow down the operation of the super buffer. These include: high circuit operating temperatures, low supply voltage, low gate voltage (which is inherent to low supply voltage in fully restored static CMOS logic), and minimum current SPICE fabrication parameters (obtained from the silicon foundry where the circuit is to be constructed). A circuit designed to function properly under these punishing conditions will function faster in a less severe environment. Of course, a circuit may be designed to meet less hostile conditions if it is not necessary for the circuit to meet the design performance standards under such extreme circumstances. In that case, nominal values for the supply voltage and SPICE fabrication parameters might be used in the design specifications as well as some lesser operating temperature.

In the design of microchips intended for commercial use, semiconductor manufacturers typically guarantee their devices for operation over a temperature range of $-40°$ C to $+85°$ C and over a supply voltage range of 4.5 V to 6 V. [Signetics Corp., 1986, p. 5-4]. For military applications both of these ranges are broader. In keeping with industrial practice, the super buffer is designed for guaranteed operation within a specified timing constraint at 85° C (358.15 K) with a supply voltage (and gate voltage) of 4.5 V. The minimum current SPICE fabrication parameters chosen are from the Metal-Oxide-Semiconductor Implementation Service (MOSIS) sponsored by DARPA at the Information Sciences Institute, University of Southern California. The parameters are included in Appendix B. The desired rise time and fall time at the output of each stage of the super buffer is chosen to be 2.0 nanoseconds. The total average delay time desired is therefore (from equation (2.16)):

$$
\begin{aligned}
t_{d_{avg,total}} &= t_{d_{avg,input\ stage}} + t_{d_{avg,output\ stage}} \\
&= \tfrac{1}{4}\cdot\left(t_{r_{input\ stage}} + t_{f_{input\ stage}}\right) + \tfrac{1}{4}\cdot\left(t_{r_{output\ stage}} + t_{f_{output\ stage}}\right) \\
&= \tfrac{1}{4}\cdot(2.0\times10^{-9}+2.0\times10^{-9}) + \tfrac{1}{4}\cdot(2.0\times10^{-9}+2.0\times10^{-9}) \\
&= 2.0\times10^{-9}[s]
\end{aligned}
$$

B. DESIGN OF THE SUPER BUFFER.

The number of registers across the width of the data path in Figure 3.1 is equal to the number of bits in the word that propagates along the path. In order to determine the loading that the super buffer must be designed to support, an upper limit on the number of registers that can be supported under a worst-case scenario must be chosen. As this number becomes larger, the size of the super buffer required also

becomes larger. Due to the size constraint imposed by the physical layout of all floorplans designed by the Monterey Silicon Compiler, the number of supportable registers under a worst-case scenario is chosen to be four. For a number larger than that the static CMOS super buffer would have to be significantly larger than the NMOS super buffer that it replaces and it simply would not fit within the space presently allotted for it on the floorplan. This, however, is the number of registers that can be driven in 2 nanoseconds if the worst-case scenario is assumed. As discussed in Section A, the worst-case scenario implies that the circuit is constructed using worst-case minimum current SPICE fabrication parameters, that the supply voltage is only 4.5 volts, and that the circuit operates at 85° C. Since the super buffer is designed under a worst-case assumption, operating it under less hostile conditions will enhance its performance. If the capacitive loading on the super buffer remains constant but operating conditions improve, the super buffer will function faster than the 2 nanosecond delay time upper limit. The actual speed can be calculated using equation (2.56). Conversely, improved operating conditions offer the opportunity to drive more capacitance (more registers) without affecting the speed of operation of the super buffer. If the worst-case scenario does occur and the data path is wider than four bits then the super buffer will, of course, still function, but it will no longer meet the 2 nanosecond design goal. The performance of the super buffer under the loading caused by four, eight, sixteen, and thirty-two bit data paths is discussed in Section E. Thus, the Monterey Silicon Compiler is not by any means restricted to the construction of four bit circuits. As with any integrated circuit, the maximum clock frequency at which the super buffers can operate is governed by temperature, capacitive loading, supply voltage, and fabrication process. The better these conditions are, the faster the super buffers will operate; the worse they are, the slower the super buffers will operate.

Figure 3.2 shows the mask level layout of a single data register [Mullarky, 1987, p. 56]. The labels PHI1! and PHI2! on Figure 3.2 correspond to the phi1 and phi2 clocklines in Figure 3.1. The output stage of one super buffer of a super buffer pair is connected to the input labeled PHI1! in Figure 3.2. The output stage of the other super buffer in the pair is connected to the input labeled PHI2! in Figure 3.2. Looking into the register at PHI1! the super buffer sees three n MOSFETs and a p MOSFET, all in parallel and each having mask level channel dimensions ($L \times W$) equal to 3.0 microns by 4.5 microns.

The programs in Appendix A are used to calculate the gate capacitance seen by the super buffer looking into the PHI1! input of the register. To run the programs on the HP41CX they must all first be stored in extended memory and main memory must be cleared. The name "VLSI" is stored in the alpha register and execution continues as follows:

GETP then R/S are executed. The program responds:

STOP. PURGE MAIN MEMORY

The programs are so large that they cannot all be held in main memory at once. They must be swapped in and out of extended memory as needed. The prompt is saying that the program VLSI should be the only program currently in main memory. If it is not, all programs in main including VLSI must be purged and the process started over again. R/S is executed again. The program responds:

ENTER FAB. PARAM. CARD

The programs require that the SPICE fabrication parameters be entered. These should be stored on magnetic cards in accordance with the data register allocation detailed in Appendix A. The cards are entered and the program responds:

OPERATING TEMP ⟨ KELVIN ⟩ = ?

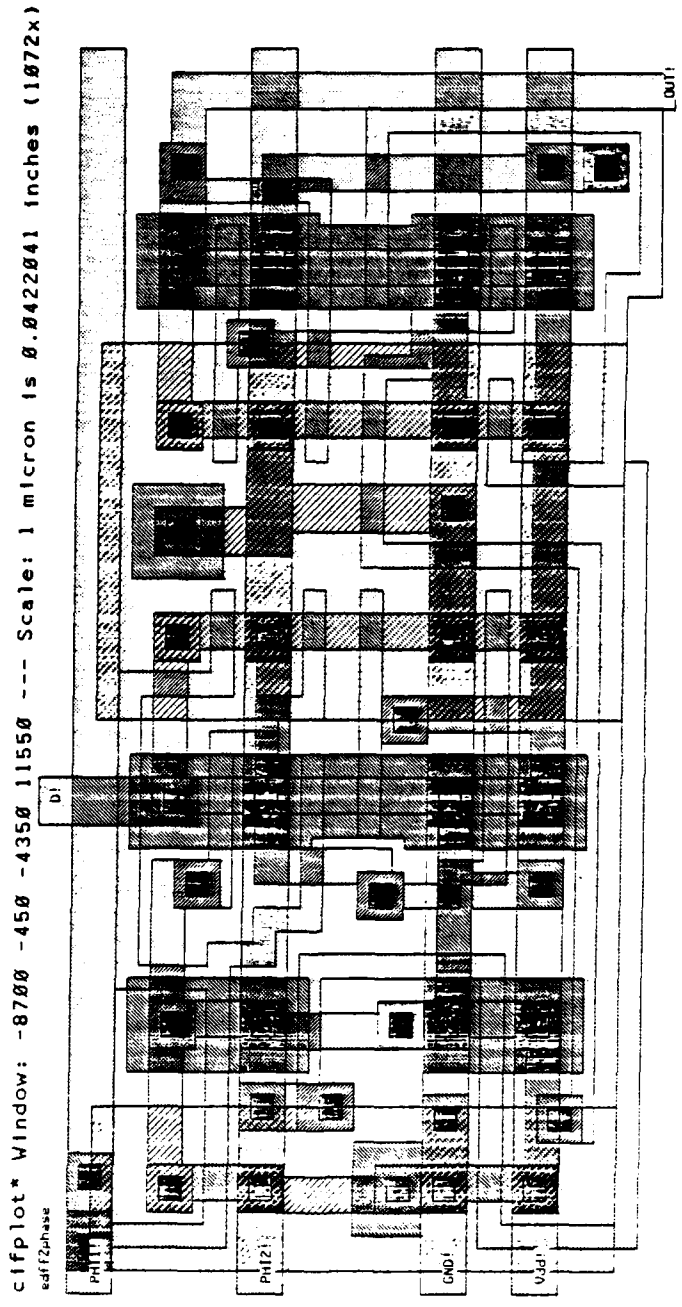


Figure 3.2 Mask Level Integrated Circuit Layout of CMOS Register

The response for worst-case operating temperature is 358.15. R/S is executed. The program responds:

VBS ⟨ N ⟩, VOLTS = ?

Since a p^+ substrate contact will be placed on the actual device so that it abuts the n source diffusion at the ground line contact, Vbs_n is equal to zero. R/S is executed. The program responds:

VBS ⟨ P ⟩, VOLTS = ?

Since an n^+ substrate contact will be placed on the actual device so that it abuts the p source diffusion at the supply voltage line contact, Vbs_p is equal to zero. R/S is executed. The program responds:

SUPPLY VOLTAGE ⟨ VOLTS ⟩ = ?

The response for worst-case supply voltage is 4.5. R/S is executed. The program responds:

P CHNL MASK LNGTH ⟨ MTR ⟩ = ?

At the end of Section B.2 in Chapter II it is stated that the channel mask length should be chosen to be the minimum feature size for the technology being used. Since the technology being used in the Monterey Silicon Compiler is based on a p well process with three micron minimum feature size the response is 3.0×10^{-6}. R/S is executed. The program responds:

N CHNL MASK LNGTH ⟨ MTR ⟩ = ?

Again, the response is 3×10^{-6}. R/S is executed. The program responds with a menu:

WI LD

Choosing WI would go into the transistor sizing algorithm. Since the immediate goal is to find the gate capacitance seen by the super buffer looking into the PHI1!

input of the register, LD is chosen, where LD is short for "load". The program responds with an introductory line immediately followed by a query:

LOADCAP

LOAD CHNL MASK LNGTH ⟨ MTR ⟩ ?

The total gate capacitance looking in at PHI1! of the register is calculated starting with the three n MOSFETs. The length for all three is 3.0 microns. R/S is executed. The program responds:

LOAD CHNL MASK WIDTH ⟨ MTR ⟩ ?

The width for all three n MOSFETs is 4.5 microns. R/S is executed. The program responds with a query followed by a menu:

DEVICE TYPE?

N P

N is chosen to indicate to the program that the device is an n MOSFET.

The program responds with another query followed by another menu:

ANOTHER DEVICE?

Y N

The response is Y for yes. The program responds.

LOAD CHNL MASK LNGTH ⟨ MTR ⟩ ?

The same procedure is continued by responding to the prompts as appropriate until all three n MOSFETs and the p MOSFET have been entered. At that point the response to the query, "ANOTHER DEVICE", is N for no, and the program responds with the total capacitance of the four MOSFETs connected in parallel:

C Σ LOAD DEV = $43.43E-15$ ⟨ FARAD ⟩

R/S is executed and the menu that gives a choice between the transistor sizing algorithm and the load capacitance algorithm appears again. Looking into the register at PHI2! the other super buffer in the pair sees three p MOSFETs and an

n MOSFET, all in parallel and each having mask level channel dimensions (L x W) equal to 3.0 microns by 4.5 microns. In terms of the number of each type of device this is the opposite of what is seen by the other super buffer. This leads to a slight difference between the gate capacitance seen by each super buffer of the pair. The load capacitance algorithm determines the gate capacitance seen looking into the register at PHI2! to be 39.5 femtofarads. The difference between the capacitance seen by the two loads is not so great as to warrant the construction of two entirely different super buffers to make up the pair. It does however suggest that if only one basic super buffer is to be designed and employed in pairs then the basic super buffer should be designed to support the larger of the two loads, 43.43 femtofarads. For each super buffer to drive four registers requires that it be capable of driving four times this amount. Thus, $C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate}$ for each super buffer $= (4)\cdot(43.43\times10^{-15}) = 173.7\times10^{-15}$ Farads. There is one other element of capacitive loading not yet determined that is required in the calculations; the routing capacitance. For a four bit data path the Monterey Silicon Compiler typically places the registers so that the metal line connecting the clock input on each register to the super buffer output is 187.5 microns long per register and 4.5 microns wide. Thus, for a four bit data path the routing capacitance of the line extending from each super buffer in the pair to the four registers it supports is:

$$C_r = (4)\cdot(4.5\times10^{-6})\cdot(187.5\times10^{-6})\cdot Cmd$$

where Cmd is the capacitance formed between metal and substrate. Cmd is taken to be $100\times10^{-6}\left[\frac{F}{m^2}\right]$ [Weste and Eshraghian, 1985, p. 135]. Substituting this value for Cmd gives:

$$C_r = 337.5 \text{ femtofarads}$$

Armed with the knowledge that $C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate} = 173.7 \times 10^{-15}$ Farads, and that $C_r = 337.5 \times 10^{-15}$ Farads for each super buffer the transistor sizing algorithm may now be used to determine the dimensions of the p and n MOSFETs in the output stage of the super buffer.

Returning once again to the menu that gives a choice between the transistor sizing algorithm and the load capacitance algorithm the choice "WI" is selected. The program responds with a query followed by a menu:

DO YOU KNOW C Σ LOAD DEV?

YES NO

Since $C\Sigma$ LOAD DEV ($= C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate}$) has already been calculated the response is YES. The program responds:

$C\Sigma$ LOAD DEV ⟨ FARAD ⟩ = ?

The response is 173.7×10^{-15}. R/S is executed and the program responds:

DESIRED RISE TIME ⟨ SEC ⟩ = ?

The time that has been chosen is 2×10^{-9} seconds. R/S is executed. The program responds:

ROUTING CAP ⟨ FARAD ⟩ = ?

C_r has been calculated to be 337.5×10^{-15} Farads. It is interesting to note that this routing capacitance is about double the gate capacitance of the load. R/S is executed. The program responds:

P DRAIN MASK LNGTH ⟨ MTR) ?

The minimum drain length for the technology is selected in order to reduce drain diffusion capacitance to a minimum. The value is 3×10^{-6}. R/S is executed. The program responds:

N DRAIN MASK LNGTH ⟨ MTR ⟩ ?

As with the p drain diffusion length, the value selected is the minimum permitted. The value is 3×10^{-6}. R/S is executed. The program responds:

NO. P DIFFUSION CONTACTS

The program needs to know how many diffusion to metal contacts are going to be placed around the p drain diffusion area. The choice is three. R/S is executed and the program responds:

P DIFCTC MASK LNGTH ⟨ MTR ⟩ ?

The value requested is the length of the diffusion to metal contacts placed around the p drain diffusion. To minimize drain capacitance the minimum dimension supported by the technology is chosen. The value is 6×10^{-6}. This is the same value for the length of the diffusion to metal contacts placed around the n drain diffusion as well as for the width of the contacts placed around both drain diffusions. R/S is executed and the program responds:

PDIFCTC MASK WIDTH ⟨ MTR ⟩ ?

The response is 6×10^{-6}. R/S is executed and the program responds:

NO. N DIFFUSION CONTACTS

One diffusion to metal contact will be placed on the n drain diffusion area. R/S is executed. The program responds:

N DIFCTC MASK LNGTH ⟨ MTR ⟩ ?

The response is 6×10^{-6}. R/S is executed. The program responds:

N DIFCTC MASK WIDTH ⟨ MTR ⟩ ?

The response is 6×10^{-6}. R/S is executed. The program responds, with all the dimensions (as defined in Table 2.2) of the n and p MOSFETs in the output stage of the super buffer that are required for a complete modeling of the output stage with PSpice:

$L \langle P \rangle = 3 \times 10^{-6} \langle MTR \rangle$

$W \langle P \rangle = 134.3 \times 10^{-6} \langle MTR \rangle$

$AD \langle P \rangle = 660.9 \times 10^{-12} \langle SQ.\ MTR \rangle$

$PD \langle P \rangle = 314.4 \times 10^{-6} \langle MTR \rangle$

$RD \langle P \rangle = 22.16 \langle OHM \rangle$

$L \langle N \rangle = 3 \times 10^{-6} \langle MTR \rangle$

$W \langle N \rangle = 51.93 \times 10^{-6} \langle MTR \rangle$

$AD \langle N \rangle = 231.2 \times 10^{-12} \langle SQ.\ MTR \rangle$

$PD \langle N \rangle = 124.4 \times 10^{-6} \langle MTR \rangle$

$RD \langle N \rangle = 29.19 \langle OHM \rangle$

The program then returns to the very beginning. The program is run again in its entirety to determine the dimensions of the n and p MOSFETs in the input stage of the super buffer. $C \sum_{\substack{\text{load} \\ \text{devices}}} \text{gate}$ for the input stage is that capacitance that is due to the gates of the n and p MOSFETs of the output stage. C_r is negligible for the input stage as the extent of the routing between the input and output stages is minimal. The drain diffusion length and contact dimensions are again are chosen to be the minimum supportable by the technology to minimize the capacitance that must be driven. Two metal to diffusion contacts are chosen for the p drain diffusion area and one is chosen for the n drain diffusion area. The results of factoring these variables into the program in combination with the worst-case temperature, supply voltage, and fabrication parameters are all the dimensions (as defined in Table 2.2) of the n and p MOSFETs in the input stage of the super buffer that are required for a <u>complete</u> modeling of the input stage with PSpice:

$L \langle P \rangle = 3 \times 10^{-6} \langle MTR \rangle$

$W \langle P \rangle = 106.1 \times 10^{-6} \langle MTR \rangle$

$$AD\ \langle P \rangle = 507.3 \times 10^{-12}\ \langle SQ.\ MTR \rangle$$

$$PD\ \langle P \rangle = 246 \times 10^{-6}\ \langle MTR \rangle$$

$$RD\ \langle P \rangle = 32.76\ \langle OHM \rangle$$

$$L\ \langle N \rangle = 3 \times 10^{-6}\ \langle MTR \rangle$$

$$W\ \langle N \rangle = 41.01 \times 10^{-6}\ \langle MTR \rangle$$

$$AD\ \langle N \rangle = 191.4 \times 10^{-12}\ \langle SQ.\ MTR \rangle$$

$$PD\ \langle N \rangle = 102.6 \times 10^{-6}\ \langle MTR \rangle$$

$$RD\ \langle N \rangle = 29.73\ \langle OHM \rangle$$

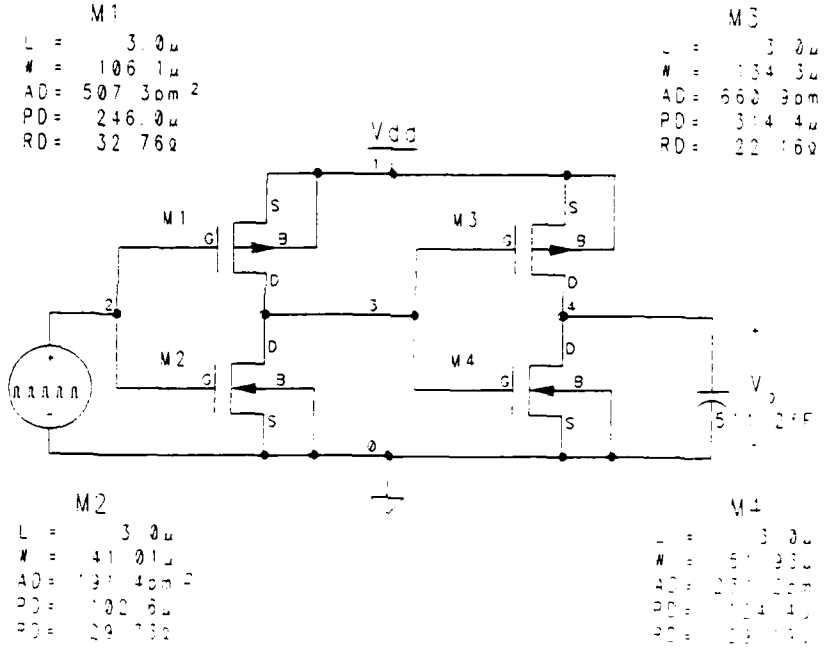


Figure 3.3 Schematic of Static CMOS Super Buffer

A schematic drawing of the super buffer with all the calculated dimensions affixed is presented in Figure 3.3. The load capacitance on the output stage is merely the sum of $C\sum_{\substack{\text{load}\\\text{devices}}}\text{gate} + C_r$ calculated for the output stage. Note that node numbers (0,1, etc.) and transistor names (M1, M2, etc.) have been assigned in the figure. These node numbers and transistor names correspond to the nodes used in the actual PSpice deck as it appears, ready for simulation, in Appendix B. The worst-case, minimum current SPICE fabrication parameters are also included in the PSpice deck in Appendix B.

C. SIMULATION OF THE SUPER BUFFER.

The results of simulating the PSpice deck in Appendix B are displayed in Figures 3.4, 3.5, and 3.6. In each figure voltage is plotted as a function of time. The voltages displayed are referred to by node numbers which correspond to those used in Figure 3.3 and Appendix B. Voltages are measured with respect to two nodes. Where only one node is listed, the second is taken to be node 0, which in the case of this circuit is ground. Thus, Figure 3.4 shows the voltage waveforms presented to and produced by the input stage of the super buffer. The value of 1.0 volt that is subtracted from the input voltage wave form is the threshold voltage of the n device. Similarly the value of $|-1.0|$ volt that is subtracted from the absolute value of V(2,1) is the absolute value of the threshold voltage of the p device. These waveforms are included to assist in determining when the devices are in their various regions of operation: linear, saturation, and cutoff as discussed in Chapter II.

The simulation result in Figure 3.4 is almost identical to that predicted by theory in Figures 2.9a and 2.10a (the small spikes are due to anomalies of the model). This is an important result because the model used to prepare Figures

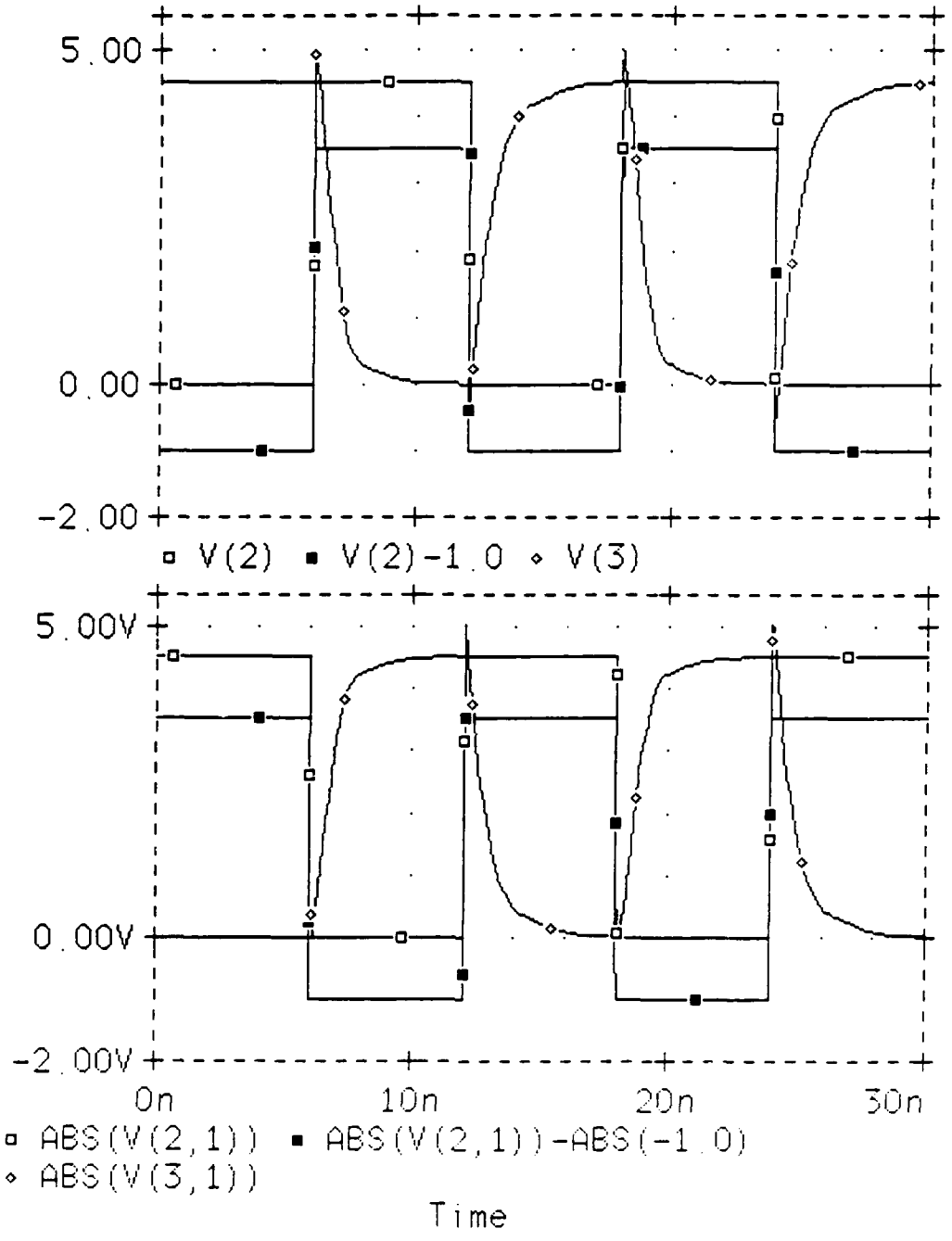


Figure 3.4 PSpice PROBE Postprocessor Simulation of Voltage Waveforms Presented to and Produced by the Input Stage of the Super Buffer Using MOSFET Level Two Model

2.9a and 2.10a was based on the Shichman-Hodges model associated with level one SPICE MOSFET simulations while the model used in the simulation that resulted in Figures 3.4, 3.5, and 3.6 was a different one. The foundation of the equations developed in Chapter II rests on the Shichman-Hodges model of MOSFET behavior. To confirm the validity of the approach taken in developing the equations of Chapter II a totally independent model was selected for simulation; the level two SPICE MOSFET model based on the work of Vladimirescu and Liu [Electronics Research Laboratory, 1980, pp. 1–23].

Figure 3.5 shows the voltage waveforms presented to and produced by the output stage of the super buffer. Figure 3.6 shows the voltage waveform presented to the input stage of the super buffer and the voltage waveform produced by the output stage of the super buffer.

Table 3.1 summarizes the rise and fall time data presented in Figures 3.4, 3.5, and 3.6. Since the performance target is that t_r and t_f for the output voltage waveforms of both the input stage and the output stage not exceed 2×10^{-9} seconds the table shows that success has been achieved. The rise time of the voltage waveform produced by the input stage is 12% faster than the 2 nanosecond goal. The fall time of this voltage waveform is 37% faster than the goal. The rise time of the voltage waveform produced by the output stage is 6% faster than the two nanosecond goal. The fall time of this waveform is 19.5% faster than the goal. Rough symmetry is achieved in rise and fall times at the output of each stage.

Delay time based on the projected rise and fall times should be 1 nanosecond for each stage or 2 nanoseconds for both stages in cascade (based on equation (2.16)). Table 3.2 summarizes the delay time data of Figures 3.4, 3.5, and 3.6. Table 3.2 shows that the average time delay through the input stage is 25% faster than predicted. The average time delay through the output stage is 27% slower

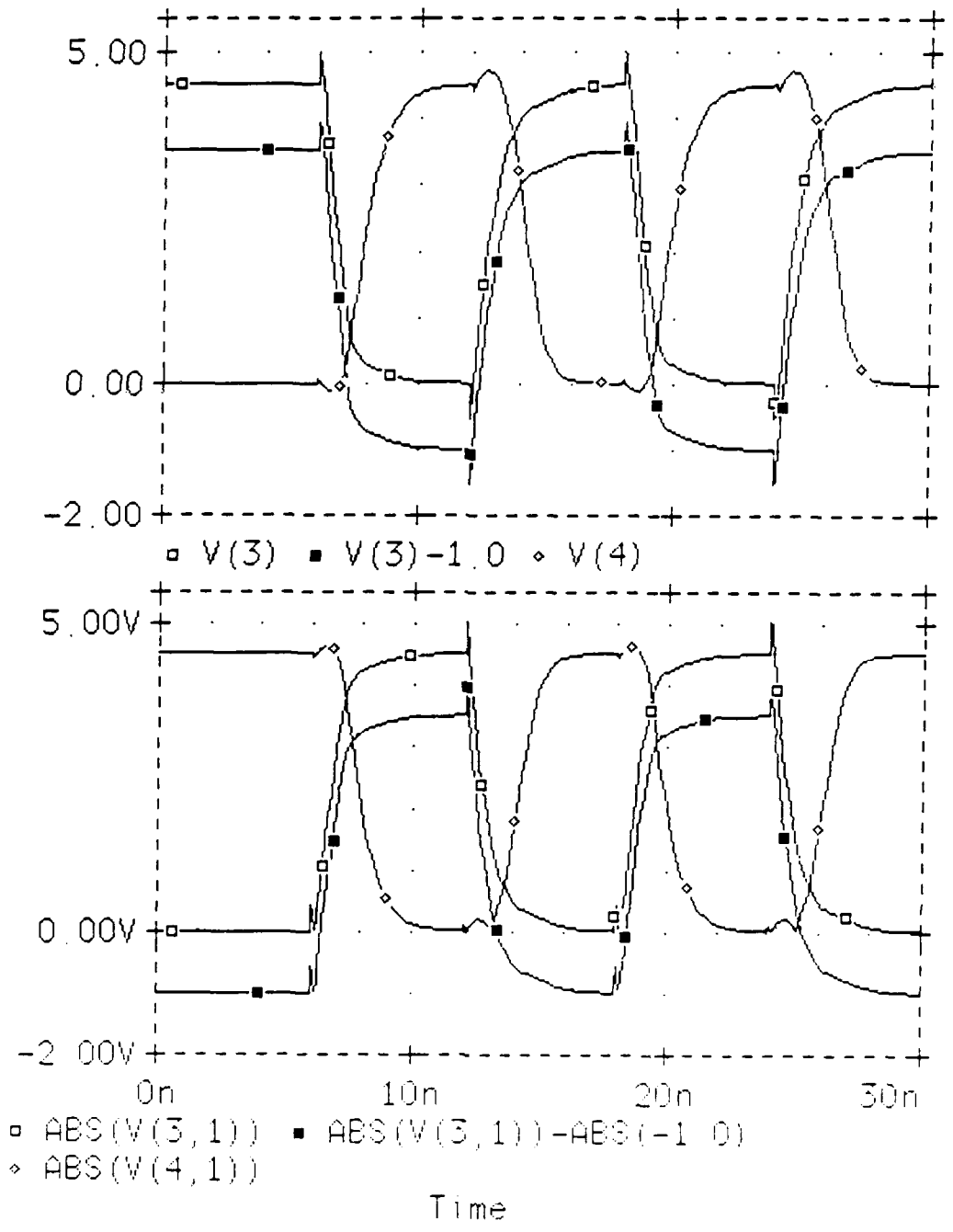


Figure 3.5 PSpice PROBE Postprocessor Simulation of Voltage Waveforms Presented to and Produced by the Output Stage of the Super Buffer Using MOSFET Level Two Model.

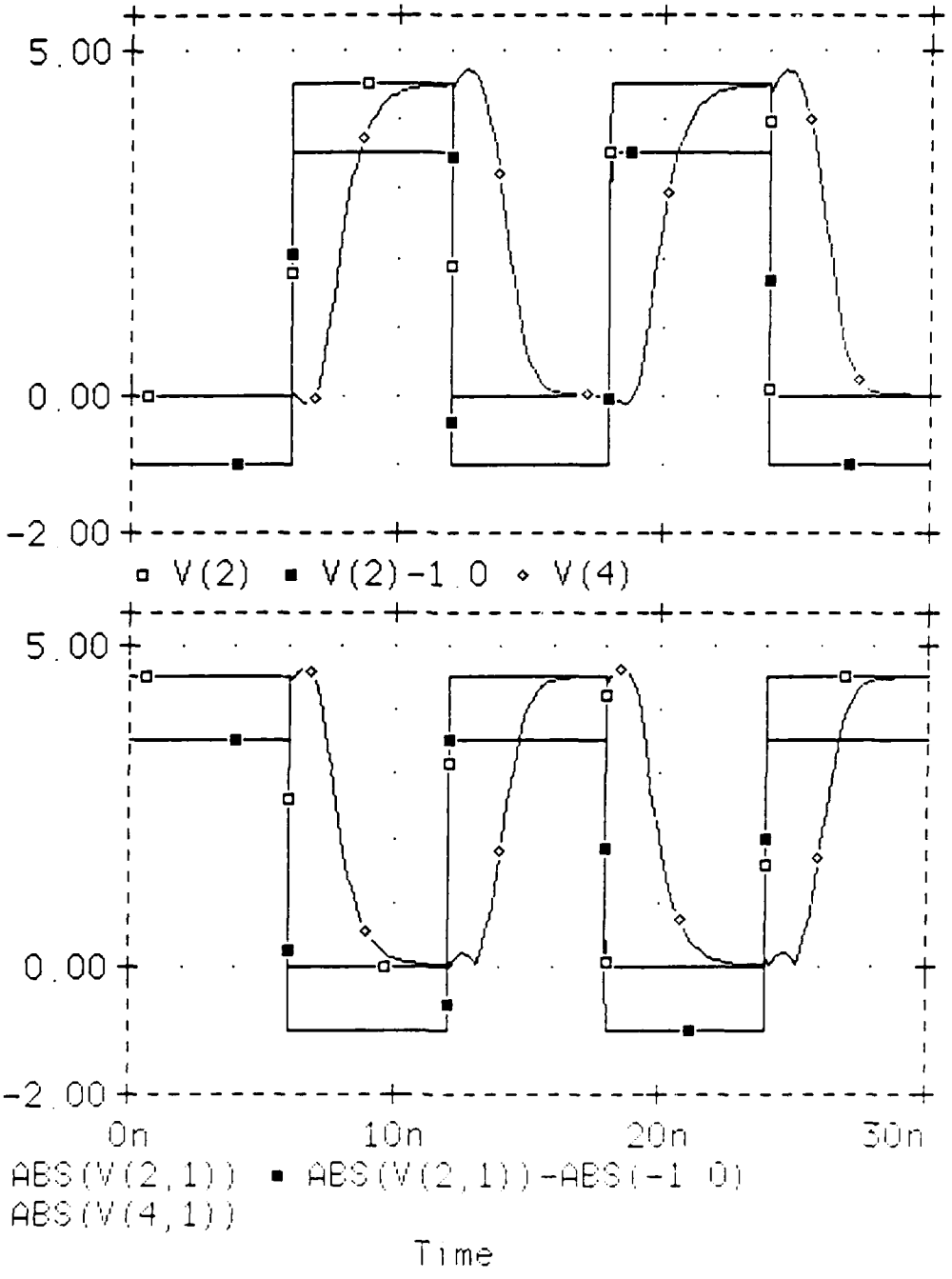


Figure 3.6 PSpice PROBE Postprocessor Simulation of Voltage Waveforms Presented to the Input Stage of the Super Buffer and Produced by the Output Stage of the Super Buffer Using MOSFET Level Two Model.

TABLE 3.1 SUMMARY OF RISE AND FALL TIME DATA FOR THE SUPER BUFFER

	$t_r(seconds)$	$t_f(seconds)$
voltage waveform at input to input stage	8×10^{-12}	10×10^{-12}
voltage waveform at output of input stage (same as at input to the output stage	1.76×10^{-9}	1.259×10^{-9}
voltage waveform at output of output stage	1.877×10^{-9}	1.61×10^{-9}

than predicted. Finally, the average time delay through the entire super buffer is within 1% of the projected value of two nanoseconds. Note that in Table 3.2 the sum of the column entries do not sum exactly to the values in the third row. This is because all the data in the table is taken directly from discrete data points that are used to draw the voltage waveforms in Figures 3.4, 3.5, and 3.6. Some linear interpolation is necessary in order to obtain the time at the 50% voltage levels. This accounts for the slight differences between the sums of the column entries and the third row entries.

Thus the correct timing is achieved and the validity of the equations in Chapter II is borne out by simulation with an independent model.

The low noise margin and high noise margin for both stages given a 5 volt supply voltage is calculated to be 2.6 volts and 1.7 volts, respectively [Weste & Eshraghian, 1985, pp. 507–508].

D. MASK LEVEL SUPER BUFFER IMPLEMENTATION

A mask level integrated circuit layout of the super buffer based on the dimensions calculated is shown in Figure 3.7. To reduce the possibility of latch-up,

TABLE 3.2 SUMMARY OF DELAY TIME DATA FOR THE SUPER BUFFER

	$t_{d_r}(seconds)$	$t_{d_f}(seconds)$	$t_{d_{avg}} = \frac{t_{d_r}+t_{d_f}}{2}$
delay through input stage	0.72×10^{-9}	0.773×10^{-9}	0.746×10^{-9}
delay through output stage	1.093×10^{-9}	1.45×10^{-9}	1.272×10^{-9}
delay through input and output stages in cascade	1.866×10^{-9}	2.17×10^{-9}	2.018×10^{-9}

substrate contacts are employed in the n^+ and p^+ source diffusion regions. To minimize the chip area of the super buffer requires that the transistors be laid out in serpentine patterns. As a final check of the circuit's functionality, the mask level design is simulated with the ESIM gate level simulator. The simulation indicates that the mask layout is logically correct.

To use the super buffer layout in the automated generation of custom microchips with the Monterey Silicon Compiler, the Caltech Intermediate Form (CIF) representation of the circuit is translated into an L5 form representation using a CIF to L5 conversion program created by E. Malagon [Malagon, 1987, p. 110]. CIF and L5 are two different languages used to represent the geometric shapes that comprise a mask level layout. L5 stands for Lincoln Laboratory Lisp-based Layout Language. The language was created at the Massachusetts Institute of Technology Lincoln Laboratory under the sponsorship of the US Defense Advanced Research Projects Agency. The Monterey Silicon Compiler requires that the mask level integrated circuit layout be represented in L5 form before installing the layout representation into the compiler code. After the design is translated from CIF to

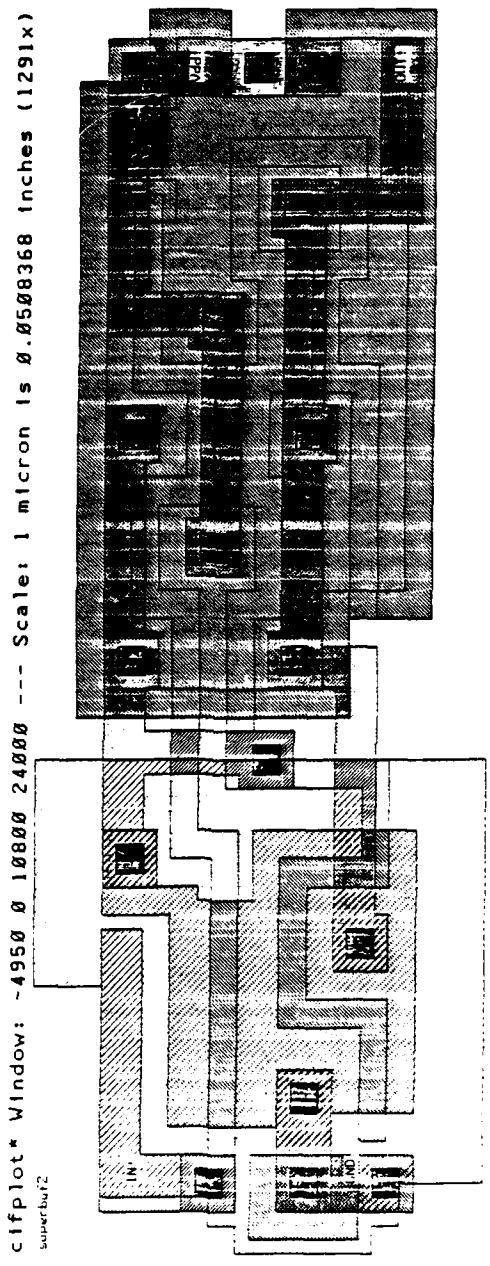


Figure 3.7 Mask Level Integrated Circuit Layout of Static CMOS Super Buffer.

L5 it is inserted into the compiler code within the program data-path.l using a procedure proposed by Baumstarck [Baumstarck, 1987, pp. 70–81]. Some additional wiring is added in L5 code to ensure that the super buffer cell is correctly placed on the floorplan each time it is called by the compiler and to ensure that signals are properly routed to and from the super buffer.

An example of a portion of a microchip designed using the replacement static CMOS super buffer is presented in Figure 3.8. In the figure the horizontal rails at the center of the figure are the supply voltage, ground, and clock distribution rails. The outputs at the top of the two super buffers that make up the super buffer pair that straddles the rails are wired to PHI1! and PHI2! of the register directly above it in the data path.

Figure 3.9 shows an entire 4 bit shifter microchip designed with the Monterey Silicon Compiler using the static CMOS super buffer and CMOS register. This chip is in fact a hybrid chip in that both NMOS and CMOS technologies are incorporated in the chip. The input and output pads around the outer ring of the circuit are NMOS. A collection of CMOS pads for the Monterey Silicon Compiler has been obtained but has not yet been translated into L5 and inserted into the compiler code. The CMOS input pad of this collection is the one that has been referred to in this chapter. It is presented in Figure 3.10.

E. PERFORMANCE OF THE SUPER BUFFER IN MULTI-BIT DATA PATH CIRCUITS

Section C documents the performance of the super buffer under the worst-case scenario given the loading caused by a four bit data path. In section B it is stated that improving the operating conditions or the fabrication conditions of the super buffer or both results in enhanced performance. Using equation (2.56) the performance of the super buffer can be predicted under different loading conditions.

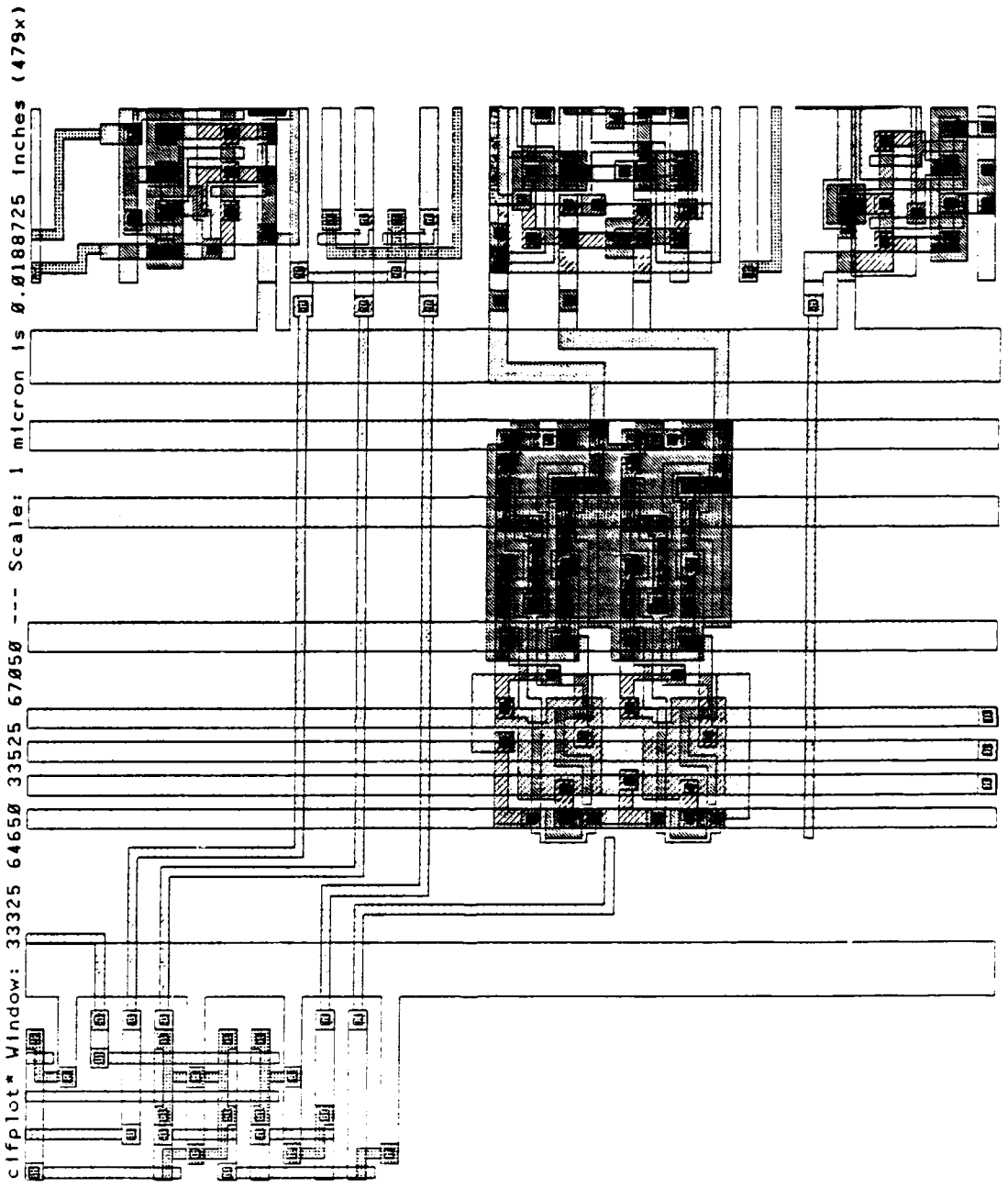


Figure 3.8 Example of a Portion of a Microchip Designed with the Monterey Silicon Compiler Showing Placement of the Static CMOS Super Buffer Below a CMOS Register.

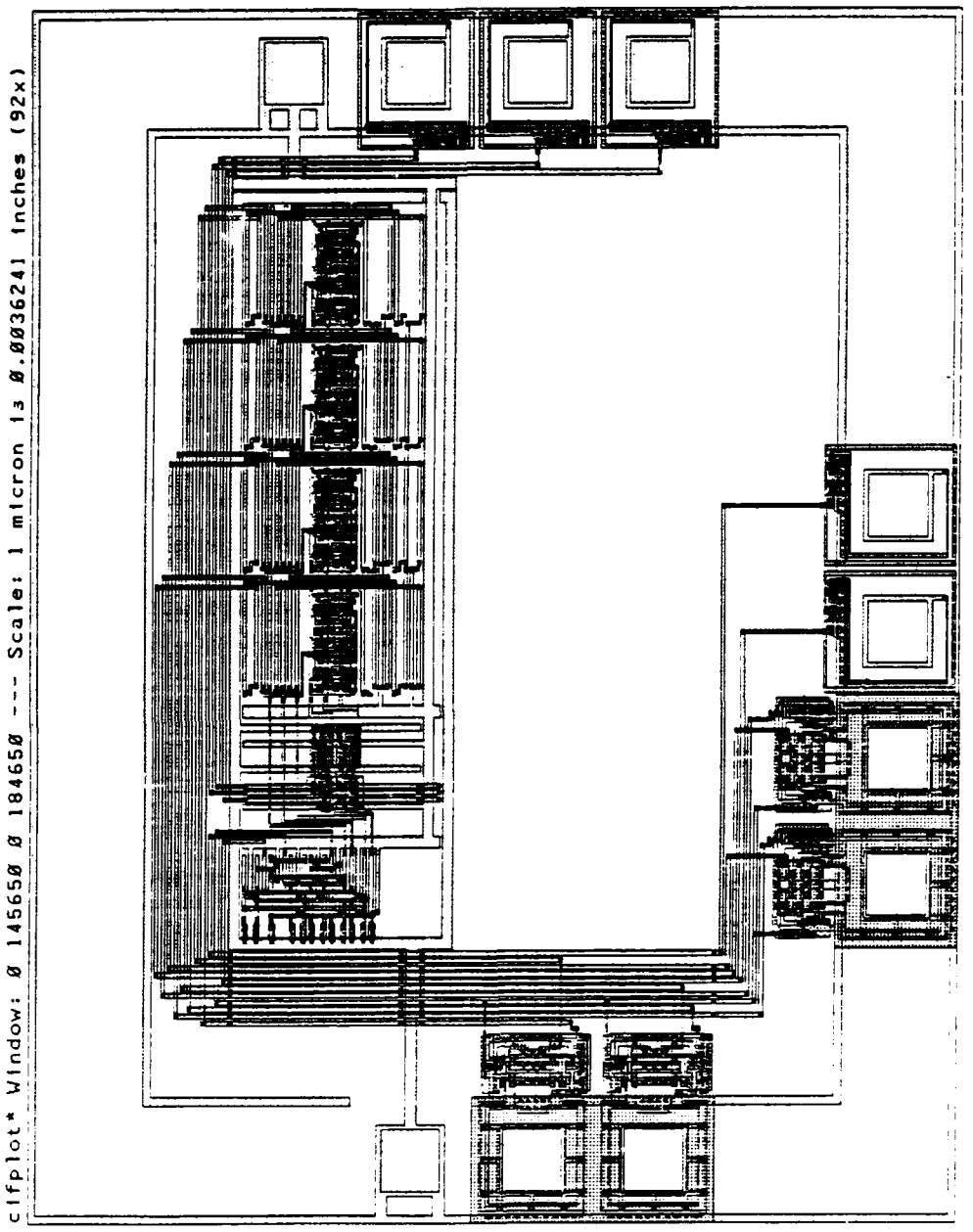


Figure 3.9 Four Bit Shifter Hybrid Microchip Designed with the Monterey Silicon Compiler.

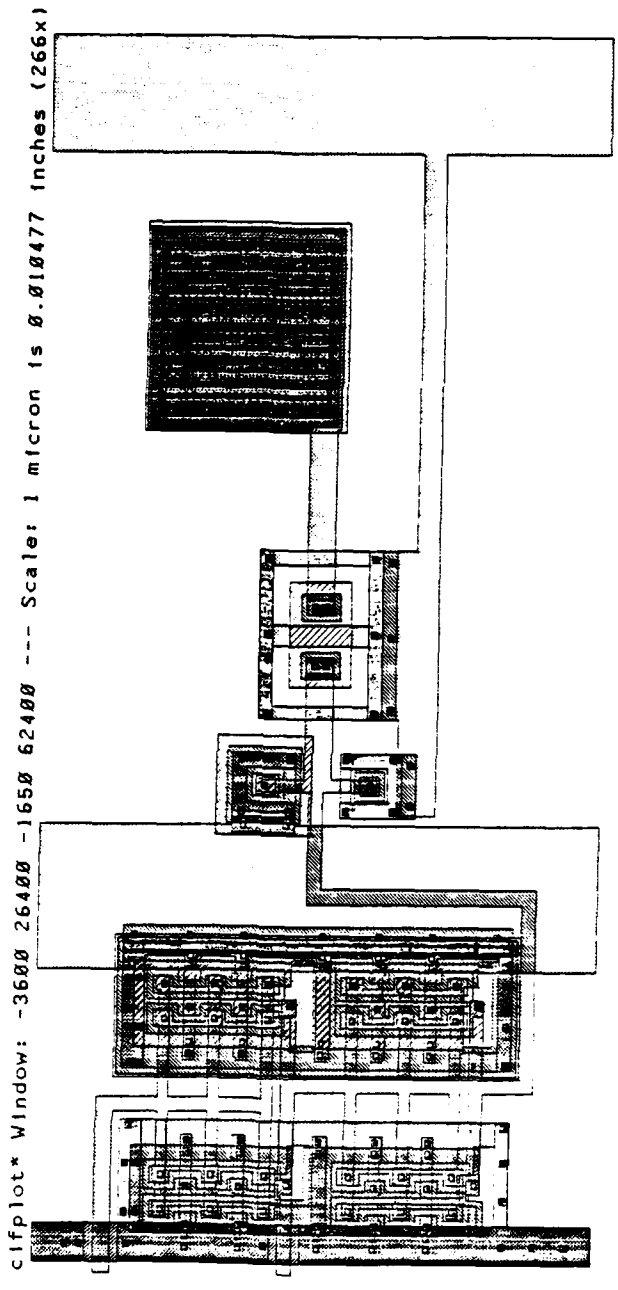


Figure 3.10 CMOS Input Pad Replacement for the Monterey Silicon Compiler.

Table 3.3 contains entries generated from equation (2.56) that predict the performance of the super buffer with the same operating temperature and fabrication process as under the worst-case scenario only the supply voltage has been raised from 4.5 volts to 6.0 volts. As can be seen in the table the effect of simply raising the supply voltage by 1.5 volts allows the super buffer to drive an eight bit data path to within 3% of the 2 nanosecond worst-case design goal.

TABLE 3.3 SUMMARY OF RISE AND FALL TIME DATA FOR THE SUPER BUFFER IN MULTI-BIT DATA PATH CIRCUITS WITH VDD = 6 VOLTS

	$t_r(=t_f)$ for 4 bits seconds	$t_r(=t_f)$ for 8 bits seconds	$t_r(=t_f)$ for 16 bits seconds	$t_r(=t_f)$ for 32 bits seconds
voltage waveform at input to input stage (ideal square wave)	0	0	0	0
voltage waveform at output of input stage (same as at input to the output stage)	1.652×10^{-9}	1.652×10^{-9}	1.652×10^{-9}	1.652×10^{-9}
voltage waveform at output of output stage	1.586×10^{-9}	2.054×10^{-9}	2.99×10^{-9}	4.862×10^{-9}

Table 3.4 contains the predicted delay time data for the super buffer under the same conditions as discussed above. The values are generated by equation (2.17) using the entries in Table 3.3. Here too, the super buffer is seen to be capable of

driving an eight bit data path in 2 nanoseconds instead of only four bits. This is due simply to the increase in the supply voltage. The table shows that even for a 32 bit data path the average delay time through the super buffer is still a very respectable 3.26 nanoseconds

TABLE 3.4 SUMMARY OF DELAY TIME DATA FOR THE SUPER BUFFER IN MULTI-BIT DATA PATH CIRCUITS WITH VDD = 6 VOLTS

	$t_{d_{avg}}$ for for 4 bits seconds	$t_{d_{avg}}$ for for 8 bits seconds	$t_{d_{avg}}$ for for 16 bits seconds	$t_{d_{avg}}$ for for 32 bits seconds
delay through input stage	0.826×10^{-9}	0.826×10^{-9}	0.826×10^{-9}	0.826×10^{-9}
delay through output stage	0.793×10^{-9}	1.027×10^{-9}	1.495×10^{-9}	2.431×10^{-9}
delay through input and output stages in cascade	1.619×10^{-9}	1.853×10^{-9}	2.321×10^{-9}	3.257×10^{-9}

Figure 3.11 shows an example of an eight bit microchip design for a taxi meter controller generated using the Monterey Silicon Compiler with the static CMOS super buffer in place [Massachusetts Institute of Technology 1982 Conference on Advanced Research in VLSI, 1982, p. 32]. Tables 3.3 and 3.4 demonstrate that by improving the supply voltage the drive of the super buffer doubles. Similar improvement in the drive of the super buffer can be realized through a fabrication process that is better than the worst-case process presumed in the design work

and through circuit operating temperatures that are lower than the worst-case temperature of 85° C. The amount of delay that is tolerable in the super buffer is largely application dependent.

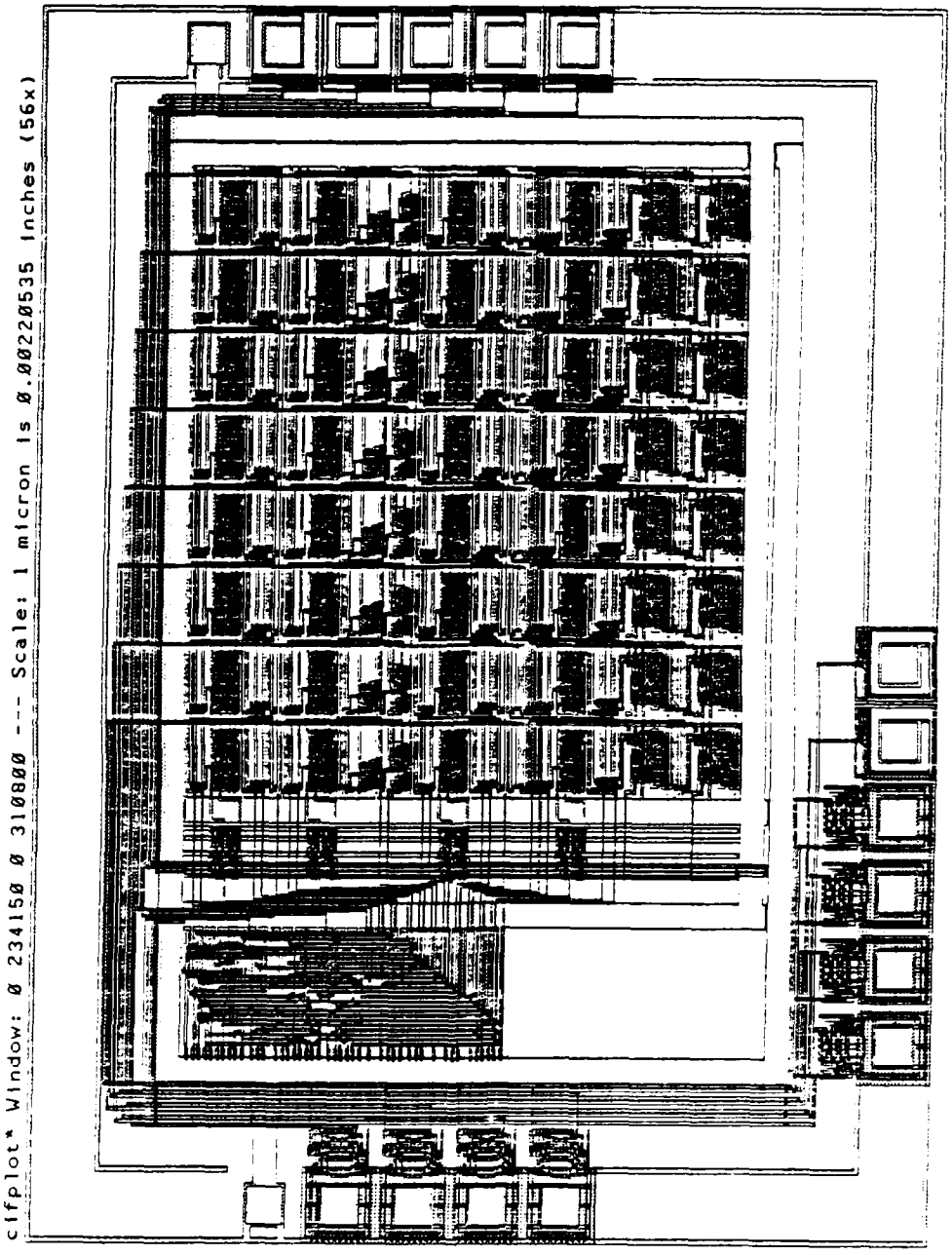


Figure 3.11 Eight Bit Taxi Meter Hybrid Microchip Designed with the Monterey Silicon Compiler.

IV. CONCLUSIONS

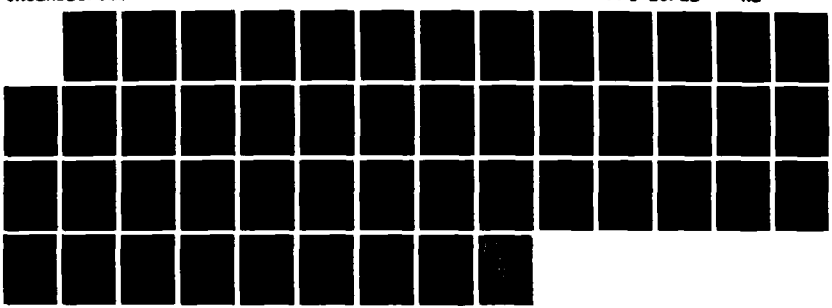
A. SUMMARY

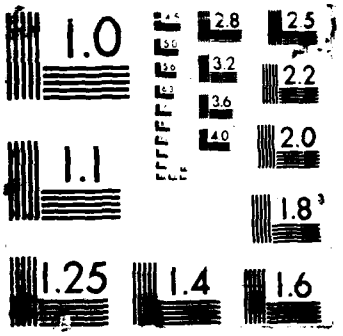
This research set out to design a high-speed static CMOS super buffer for the Monterey Silicon Compiler. The problem of choosing its dimensions to achieve a desired timing result under worst-case conditions was approached analytically, and an algorithm was derived to solve the problem. The algorithm was implemented in a computer program which was used as a computer-aided-design (CAD) tool to design a hardware device. In SPICE simulations the resulting device surpassed the pre-specified timing performance standards under the worst-case scenario that it was designed to meet. Finally, a mask level integrated circuit design of this device was installed in the Monterey Silicon Compiler as a part of the technology upgrade of the MacPITTS Silicon Compiler, where it is now available for use in the automated design of custom hybrid CMOS/NMOS VLSI microchips.

The ability to use an algorithm to size transistors on a chip is an important step toward silicon compilation that results in optimal time delay circuits because computers can think in terms of algorithms, but not in terms of rules of thumb as many engineers do. Human engineers can design, simulate, and redesign until they get timing right. A silicon compiler that tries to achieve desired timing parameters for delay, rise, and fall time has to be taught how to think about the problem. That is perhaps the most exciting application of this type of approach—compilers that design chips to meet specified timing performance standards. For future research, the Monterey Silicon Compiler code that was modified to include the static CMOS super buffer can be found in the files /a/work/steele/macpitts/data-path.l

AD-A196 526 TRANSISTOR SIZING IN THE DESIGN OF HIGH-SPEED CMOS (COMPLEMENTARY-SYMMETR..(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA G R STEELE MAR 88 2/2

UNCLASSIFIED F/G 20/12 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

and /a/work/steele/macpitts/data-path.o on the Integrated Solutions CAD workstations of the Electrical and Computer Engineering Department Computer Laboratories at the Naval Postgraduate School. The NMOS logic cells in the Monterey Silicon Compiler that still require CMOS replacements are listed in Table 4.1. This table was extracted from a similar list compiled by E. Malagon [Malagon, 1985, p.95].

TABLE 4.1 MONTEREY SILICON COMPILER NMOS LOGIC CELLS REQUIRING CMOS REPLACEMENTS

ORGANELLE	LIBRARY FUNCTIONS
layout - = organelle	=
layout - equ - organelle	word – equ
layout - <> organelle	<>
layout - <> 0 organelle	<> 0
layout - = 0 organelle	= 0
lsh - zero/ lsh2/ lsh3/ lsh4/ lsh8	≪
rsh - zero/ rsh2/ rsh3/ rsh4/ rsh8	≫
layout - odd - operand	
layout - even - operand	
port - output 1	
bit	
layout - inverting - super - buffer	
15 contacts: pc, ndc, pdc, nsc, psc	
entire controller section	

B. RECOMMENDATIONS

An algorithm for transistor sizing in static CMOS logic design has been derived and a logic device whose dimensions were determined by the algorithm has performed successfully in SPICE simulations. It is recommended that future research in this area investigate the possibility of incorporating the following topics in the algorithm: the effects of a non-ideal input square wave, empirical adjustment of the basic drain current equations, the effects of channel length modulation and

of the body effect phenomenon, accuracy in computation and in fabrication, and stage ratios.

1. Effects of a Non-Ideal Input Square Wave

The derivation of the equations for transistor sizing in a static CMOS inverter in Chapter II assumes that the input voltage waveform is a perfect square wave—one that rises and falls instantaneously. This type of waveform causes one of the MOSFETs in the inverter to be in cutoff during switching thereby permitting the circuit simplifications represented in Figures 2.9b and 2.10b. A more realistic input voltage waveform is depicted in Figure 3.5. Applying the definitions of the three regions of operation of MOSFETs presented in Table 2.1 to the input and output voltage waveforms of Figure 3.5 shows that with a non-ideal square wave applied at the input of an inverter, both transistors of the inverter will momentarily conduct during switching—one sourcing current, the other sinking current. This obviously affects circuit timing because the rate at which charge is delivered to or taken away from the load is decreased. There are two possible solutions to this in the derivation of the equations, one difficult, the other easy. The more difficult approach is to rederive the equations of Chapter II based on a time varying input voltage waveform. The easier approach is conduct a study to find an empirical adjustment factor that would modify the value presented to the algorithm as being the desired rise time to compensate for the effects of a less than ideal input square wave.

2. Empirical Adjustment of Drain Current Equations

All of the drain current equations in Table 2.1 can be multiplied by a factor determined empirically to make the drain current equations more perfectly reflect physical device behavior as suggested by Hodges and Jackson [Hodges and Jackson, 1983, pp. 51–52]. For n channel MOSFETs the factor is $(1 + \text{LAMBDA}_n \cdot Vds_n)$.

For p channel MOSFETs the factor is $(1+\text{LAMBDA}_p \cdot |Vds_p|)$. Using these factors makes it necessary to rederive the transistor sizing equations of Chapter II based on these new drain current equations.

3. Channel Length Modulation

It is known that devices with channel length shorter than 10 microns saturate before the channel pinches off because the electrons (holes) reach the scattering limited velocity in the channel [Electronics Research Laboratory, 1980, p.6]. Once in the saturation region, the channel becomes shorter than L_{eff}. To compensate, the saturation region drain current equations in Table 2.1 can be divided by a channel length modulation factor. For n channel saturated MOSFETs the factor is $(1 - \text{LAMBDA}_n \cdot Vds_n)$ [Electronics Research Laboratory, 1980, p. 15]. For p channel saturated MOSFETs the factor is $(1 - \text{LAMBDA}_p \cdot |Vds_p|)$. Making the modifications necessitates rederiving the transistor sizing equations of Chapter II based on the modified drain current equations.

4. Body Effect

In the extension of the equations derived for transistor sizing in a static CMOS inverter to general static CMOS logic design body-effect is neglected. This could perhaps be compensated for by introducing average values for Vbs_n and Vbs_p in the same manner as was used to calculate values of $Cbdj_{\text{avg}}\{T\}$ and $Cbds_{\text{avg}}\{T\}$ in equations (2.26) and (2.27).

5. Accuracy in Computation and in Fabrication

It is important to remember the first axiom of computing in using the transistor sizing algorithm presented in this thesis: "garbage in, garbage out." If for example the SPICE fabrication parameters given to the algorithm are not measured accurately, the algorithm cannot hope to produce a circuit that will behave as desired. The same is true if, for example, during fabrication the pattern

definitions created by the lithography process are inaccurate or the diffusion process is poorly controlled.

6. Stage Ratio

For consistency of approach, the input stage dimensions of the super buffer were designed using the same algorithm used to calculate the output stage dimensions. This resulted in the output stage being 1.266 times larger than the input stage. This value is known as the stage ratio which is the value by which successive transistor widths are multiplied when inverters are cascaded. Mead and Conway have written that the value of the stage ratio that minimizes overall delay for a series of cascaded inverters is given by Euler's number, $e = 2.718\cdots$ [Mead and Conway, 1908, pp. 12–14]. Thus, it may be possible to achieve slightly reduced area for the super buffer by using the algorithm to design the output stage, then using the stage ratio 2.718 to fix the dimensions of the input stage. However, Mead and Conway as well as Weste and Eshraghian have all acknowledged that other design criterion may steer a designer away from a value of 2.718 for the stage ratio [Mead and Conway, 1980, p. 14 and Weste and Eshraghian, 1895, p. 197]. Since the relative time penalty for using the stage ratio of 1.266 is slight, and since using this ratio allows a better assessment of the performance of the algorithm, the value of 1.266 was retained and the resulting circuit met its performance goals.

7. Suggested Modifications to the Monterey Silicon Compiler

In a design generated by the Monterey Silicon Compiler the area reserved for the super buffer on the floorplan is fixed. Were this area variable the compiler could be given several super buffers to choose from and it would select the one that best meets the drive requirements of the data path. This would involve modifying the body of code that defines the metal skeleton that distributes power, ground, and clock signals on chip.

The data registers currently in use are very tall and thin. Since the wires that connect the super buffer to the registers traverse the longest distance across the registers the routing capacitance has become unreasonably large. The solution is to return to the general configuration used by the original NMOS data registers. These registers were short but wide, greatly decreasing the length of the wires that connect the super buffers to the data register and thus the routing capacitance.

APPENDIX A:

STATIC CMOS INVERTER TRANSISTOR SIZING PROGRAMS

This appendix contains programs that implement the transistor sizing equations of Chapter II for a static CMOS inverter. The programs run on an HP41CX with a magnetic card reader and two extended memory modules. Main memory must be apportioned so that SIZE $\geq$ 076.

Executing the programs in certain sequences occasionally results in the message "NO ROOM" and a halt to program execution. If this occurs, the following corrective actions should be taken (in the order listed):

(1) Clear the current program from main memory.

(2) Place the name of the program that could not be loaded in the ALPHA register.

(3) Execute GETP.

(4) Execute R/S.

Program execution should continue normally with data storage registers and user flags unaffected by the temporary halt in program execution.

The interrelationships of the programs are depicted in Figure A.1. To run the programs, main memory must first be entirely cleared and all programs must be placed in extended memory. The name "VLSI" is placed in the ALPHA register, GETP is executed followed by R/S. The program VLSI is responsible for loading the SPICE fabrication parameters. These parameters must be stored on magnetic data storage cards in accordance with the data register allocation given in Table A.1. The cards are loaded when they are requested by the program. The program VLSI automatically clears itself from main memory, loads the program

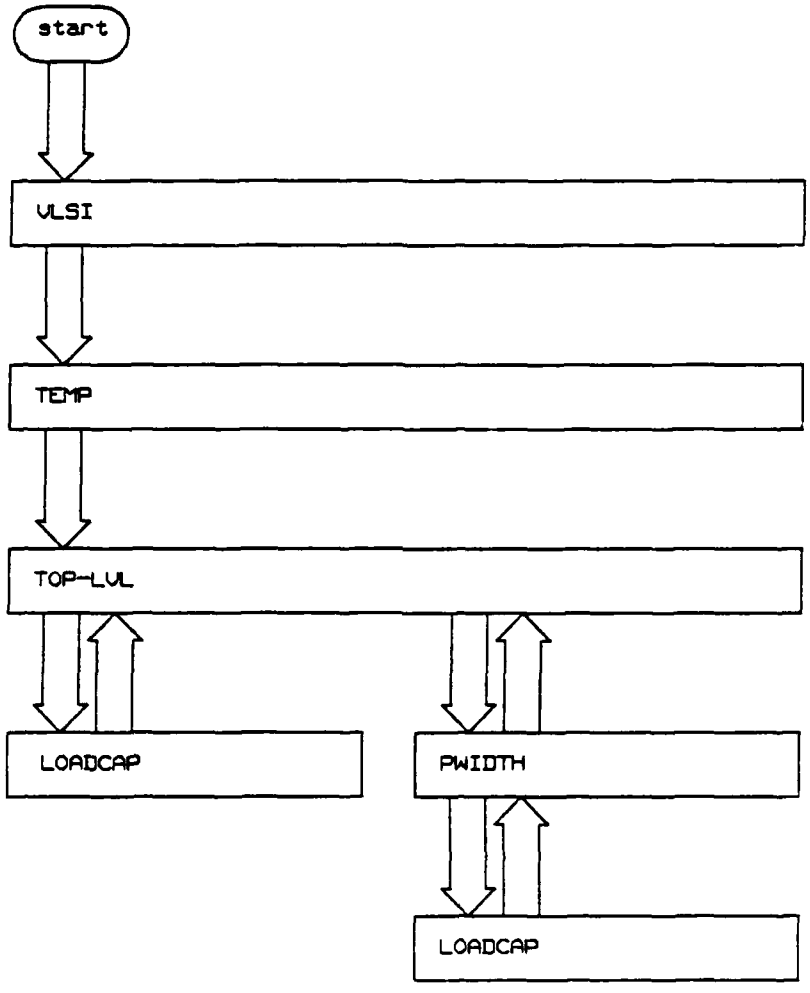


Figure A.1 Interrelationships of Transistor Sizing Programs

TEMP from extended memory, and continues execution with the first line of code in the program TEMP. The program TEMP calculates the effects of temperature on the fabrication parameters in Table A.1, overwriting the data registers of the temperature dependent parameters with their temperature altered values. Program TEMP then clears itself from main memory, loads the program TOP-LVL from extended memory, and continues execution with the first line of code in the program TOP-LVL. The program TOP-LVL acts as a memory manager, shifting the programs PWIDTH and LOADCAP in and out of main memory as required by the user. The program PWIDTH calculates all the SPICE MOSFET model dimensions and drain resistances required for the p MOSFET and the n MOSFET in a static CMOS inverter required to drive a given load capacitance. If the load capacitance is not known but the dimensions of the load MOSFETs are known, the load capacitance can be calculated using the program LOADCAP. Table A.2 details the remaining data register allocation. The source code for the programs VLSI, TEMP, TOP-LVL, PWIDTH, and LOADCAP follows Tables A.1 and A.2.

TABLE A.1 SPICE FABRICATION PARAMETER DATA REGISTER ALLOCATION

SPICE PARAMETER	DATA REGISTER FOR n DEVICES	DATA REGISTER FOR p DEVICES
KP	R_{00}	R_{17}
VTO	R_{01}	R_{18}
LD	R_{02}	R_{19}
CJ	R_{03}	R_{20}
CJSW	R_{04}	R_{21}
FC	R_{05}	R_{22}
MJ	R_{06}	R_{23}
PB	R_{07}	R_{24}
MJSW	R_{08}	R_{25}
PHI	R_{09}	R_{26}
GAMMA	R_{10}	R_{27}
$^*Cmd$	R_{11}	R_{28}
$^*\epsilon_{rel,oxide}$	R_{12}	R_{29}
CGSO	R_{13}	R_{30}
CGDO	R_{14}	R_{31}
TOX	R_{15}	R_{32}
UO	R_{16}	R_{33}
RSH	R_{73}	R_{74}

* Not SPICE parameters

TABLE A.2 DATA REGISTER ALLOCATION FOR VARIABLES OTHER THAN SPICE FABRICATION PARAMETERS

VARIABLE	DATA REGISTER FOR n DEVICES	DATA REGISTER FOR p DEVICES
T [Kelvin]	R_{34}	R_{34}
EG $\{T\}$	R_{35}	R_{35}
EG $\{Tnom\}$	R_{36}	R_{36}
Temporary	$R_{37}, R_{38}, R_{39}, R_{40}$	$R_{41}, R_{42}, R_{43}, R_{75}$
Vto $\{T\}$	R_{34}	R_{35} (abs. value)
A	R_{36}	R_{36}
Vdd	R_{38}	R_{38}
Channel Mask Length	R_{66}	R_{42}
$Cbdj_{avg}\{T\}$	R_{44}	R_{45}
$Cbds_{avg}\{t\}$	R_{46}	R_{47}
B	R_{48}	R_{48}
t_r	R_{49}	R_{49}
$E(=C_r)$	R_{50}	R_{50}
Drain Mask Length	R_{52}	R_{51}
No. Drain Diffusion Contacts	R_{56}	R_{53}
Diffusion Contact Mask Length	R_{57}	R_{54}
Diffusion Contact Mask Width	R_{58}	R_{55}
C	R_{59}	R_{59}
D	R_{60}	R_{60}
F	R_{61}	R_{61}
G	R_{62}	R_{62}
H	R_{63}	R_{63}
W	R_{65}	R_{64}
AD = AS	R_{70}	R_{67}
PD = PS	R_{71}	R_{68}
RD = RS	R_{72}	R_{69}

PROGRAM VLSI

```
01 LBL ⊤ VLSI
02 ⊤ STOP. PURGE MAI
03 ⊤ ⊢ N MEMORY
04 PROMPT
05 000.074
06 ⊤ ENTER FAB. PARA
07 ⊤ ⊢ M. CARD
08 CF 09
09 CF 10
10 AVIEW
11 RDTAX
12 ⊤ TEMP
13 GETP
14 END
```

PROGRAM TEMP

```
01          LBL ⊤TEMP
02          ⊤OPERATING TEMP ⟨
03          ⊤⊢KELVIN ⟩ = ?
04          PROMPT
05          STO 34
* * * CALCULATE EG ⟨ T ⟩ * * *
06          STO 37
07          XEQ⊤EQ⟨ T ⟩
08          RCL 37
09          STO 35
* * * CALCULATE EG ⟨ Tnom ⟩ * * *
10          300.15
11          STO 37
12          XEQ⊤EG ⟨ T ⟩
13          RCL 37
14          STO 36
* * * CALCULATE PBn ⟨ T ⟩ * * *
15          RCL 07
16          STO 40
17          STO 37
18          XEQ⊤PB ⟨ T ⟩
19          RCL 37
20          STO 07
* * * CALCULATE PBp ⟨ T ⟩ * * *
21          RCL 24
22          STO 41
23          STO 37
24          XEQ⊤PB ⟨ T ⟩
25          RCL 37
26          STO 24
* * * CALCULATE PHIn ⟨ T ⟩ * * *
27          RCL 09
28          STO 37
29          XEQ⊤PB ⟨ T ⟩
30          RCL 37
31          STO 09
* * * CALCULATE PHIp ⟨ T ⟩ * * *
32          RCL 26
33          STO 37
34          XEQ⊤PB ⟨ T ⟩
35          RCL 37
36          STO 26
* * * CALCULATE CJn ⟨ T ⟩ * * *
37          RCL 03
38          STO 37      ; CJn
39          RCL 06
40          STO 38      ; MJn
41          RCL 07
```

```
42          STO 39      ; PBn ⟨ T ⟩
43          RCL 40
44          STO 42      ; PBn
45          XEQ^TCJ ⟨ T ⟩
46          RCL 37
47          STO 03
* * * CALCULATE CJp ⟨ T ⟩ * * *
48          RCL 20
49          STO 37      ; CJp
50          RCL 23
51          STO 38      ; MJp
52          RCL 24
53          STO 39      ; PBp ⟨ T ⟩
54          RCL 41
55          STO 42      ; PBp
56          XEQ^TCJ ⟨ T ⟩
57          RCL 37
58          STO 20
* * * CALCULATE CJSWn ⟨ T ⟩ * * *
59          RCL 04
60          STO 37      ; CJSWn
61          RCL 08
62          STO 38      ; MJSWn
63          RCL 07
64          STO 39      ; PBn ⟨ T ⟩
65          RCL 40
66          STO 42      ; PBn
67          XEQ^TCJ ⟨ T ⟩
68          RCL 37
69          STO 04
* * * CALCULATE CJSWp ⟨ T ⟩ * * *
70          RCL 21
71          STO 37      ; CJSWp
72          RCL 25
73          STO 38      ; MJSWp
74          RCL 24
75          STO 39      ; PBp ⟨ T ⟩
76          RCL 41
77          STO 42      ; PBp
78          XEQ^TCJ ⟨ T ⟩
79          RCL 37
80          STO 21
* * * CALCULATE KPn ⟨ T ⟩ * * *
81          RCL 00
82          STO 37
83          XEQ^TKP ⟨ T ⟩
84          RCL 37
85          STO 00
* * * CALCULATE KPp ⟨ T ⟩ * * *
86          RCL 17
```

```
87          STO 37
88          XEQ⊤KP ⟨ T ⟩
89          RCL 37
90          STO 17
* * * CALCULATE UO_n ⟨ T ⟩ * * *
91          RCL 16
92          STO 37
93          XEQ⊤KP ⟨ T ⟩
94          RCL 37
95          STO 16
* * * CALCULATE UO_p ⟨ T ⟩ * * *
96          RCL 33
97          STO 37
98          XEQ⊤KP ⟨ T ⟩
99          RCL 37
100         STO 33
* * * CALCULATE ε_rel, oxide_n ⟨ T ⟩ * * *
101         RCL 00
102         STO 37      ; KP_n ⟨ T ⟩
103         RCL 15
104         STO 38      ; TOX_n
105         RCL 16
106         STO 39      ; UO_n ⟨ T ⟩
107         XEQ⊤EREL ⟨ T ⟩
108         RCL 37
109         STO 12
* * * CALCULATE ε_rel, oxide_p ⟨ T ⟩ * * *
110         RCL 17
111         STO 37      ; KP_p ⟨ T ⟩
112         RCL 32
113         STO 38      ; TOX_p
114         RCL 33
115         STO 39      ; UO_p ⟨ T ⟩
116         XEQ⊤EREL ⟨ T ⟩
117         RCL 37
118         STO 29
119         ⊤TOP-LVL
120         GETP
* * * EG ⟨ T ⟩ SUBROUTINE * * *
121         LBL⊤EG ⟨ T ⟩
122         RCL 37
123         X ↗ 2
124         .000702
125         *
126         RCL 37
127         1108
128         +
129         /
130         1.16
131         X ⟨ ⟩ Y
```

```
132             −
133             STO 37
134             RTN
* * * PB ⟨ T ⟩, PHI ⟨ T ⟩ SUBROUTINE * * *
135             LBL⊤PB ⟨ T ⟩
136             RCL 37
137             RCL 34
138             *
139             300.15
140             /
141             STO 38          ; Interim Result
142             3
143             ENTER ↗
144             8.6173468 E-05          ; K/q
145             *
146             RCL 34
147             *
148             RCL 34
149             300.15
150             /
151             LN
152             *
153             RCL 38
154             X ⟨ ⟩ y
155             −
156             RCL 35
157             −
158             RCL 36
159             RCL 34
160             *
161             300.15
162             /
163             +
164             STO 37
165             RTN
* * * CJ ⟨ T ⟩, CJSW ⟨ T ⟩ SUBROUTINE * * *
166             LBL⊤CJ ⟨ T ⟩
167             RCL 39
168             RCL 42
169             /
170             1
171             X ⟨ ⟩ Y
172             −
173             RCL 34
174             300.15
175             −
176             .0004
177             *
178             +
179             RCL 38
```

```
180             *
181             1
182             +
183             RCL 37
184             *
185             STO 37
186             RTN
* * * KP ⟨ T ⟩, UO⟨ T ⟩ SUBROUTINE * * *
187             LBL^T KP ⟨ T ⟩
188             RCL 34
189             300.15
190             /
191             −1.5
192             Y ↗ X
193             RCL 37
194             *
195             STO 37
196             RTN
* * * ε_rel, oxide ⟨ T ⟩ SUBROUTINE * * *
197             LBL^T EREL ⟨ T ⟩
198             RCL 37
199             100
200             *
201             RCL 38
202             *
203             RCL 39
204             /
205             8.854187818  E-14
206             /
207             STO 37
208             RTN
209             END
```

PROGRAM TOP-LVL

```
01          LBL^TOP-LVL
02          FS? 09
03          GTO 30
04          CF 02
05          CF 03
06          ENG 3
* * * CALCULATE Vto_n ⟨ T ⟩ * * *
07          RCL 01
08          STO 37      ; VTO_n
09          RCL 10
10          STO 39      ; GAMMA_n
11          RCL 09
12          STO 40      ; PHI_n ⟨ T ⟩
13          ^VBS ⟨ N ⟩, VOLTS = ?
14          PROMPT
15          ABS
16          STO 41      ; | Vbs_n |
17          XEQ^VTO ⟨ T ⟩
18          RCL 37
19          STO 34
* * * CALCULATE Vto_p ⟨ T ⟩ * * *
20          RCL 18
21          ABS
22          STO 37      ; | VTO_p|
23          RCL 27
24          STO 39      ; GAMMA_p
25          RCL 26
26          STO 40      ; PHI_p ⟨ T ⟩
27          ^VBS ⟨ P ⟩, VOLTS = ?
28          PROMPT
29          STO 41      ; Vbs_p
30          XEQ^VTO ⟨ T ⟩
31          RCL 37
32          STO 35
* * * INPUT Vdd * * *
33          ^SUPPLY VOLTAGE ⟨
34          ^⊢VOLTS ⟩ = ?
35          PROMPT
36          STO 38
* * * CALCULATE RATIO "A"; * * *
* * * W_n = W_p · A for t_r ≃ t_f      * * *
37          19
38          RCL 38
39          *
40          20
41          RCL 34
42          *
```

```
43          -
44          RCL 38
45          /
46          LN
47          2
48          /
49          STO 39      ; Interim Result
50          RCL 34
51          .1
52          RCL 38
53          *
54          -
55          RCL 38
56          RCL 34
57          -
58          /
59          RCL 39
60          +
61          RCL 38
62          RCL 35
63          -
64          *
65          RCL 17
66          *
67          STO 39      ; Interim Result
68          19
69          RCL 38
70          *
71          20
72          RCL 35
73          *
74          -
75          RCL 38
76          /
77          LN
78          2
79          /
80          STO 40      ; Interim Result
81          RCL 35
82          .1
83          RCL 38
84          *
85          -
86          RCL 38
87          RCL 35
88          -
89          /
90          RCL 40
91          +
92          RCL 38
93          RCL 34
```

```
94          −
95          *
96          RCL 00
97          *
98          STO 40      ; Interim Result
99          ᵀP CHNL MASK LNG
100         ᵀ⊢TH ⟨ MTR ⟩ = ?
101         PROMPT
102         STO 42
103         ᵀN CHNL MASK LNG
104         ᵀ⊢TH ⟨ MTR ⟩ = ?
105         PROMPT
106         STO 66
107         2
108         RCL 02
109         *
110         −
111         RCL 39
112         *
113         STO 39      ; Interim Result
114         RCL 42
115         2
116         RCL 19
117         *
118         −
119         RCL 40
120         *
121         1/X
122         RCL 39
123         *
124         STO 36
* * * MENU * * *
125         LBL 01
126         SF 27
127         ᵀWI _ _ _ _ _ _ _ _ _ LD
128         PROMPT
* * * CALL TO PWIDTH * * *
129         LBL A
130         CF 27
131         ᵀPWIDTH
132         GETP
133         GTO 01
* * * CALL TO LOADCAP * * *
134         LBL E
135         CF 03
136         CF 27
137         SF 09
138         ᵀLOADCAP
139         GETSUB
140         XEQᵀLOADCAP
141         LBL 30
```

```
142             CF 09
143             ᵀLOADCAP
144             PCLPS
145             CF 02
146             GTO 01
* * * VTO ⟨ T ⟩ SUBROUTINE * * *
147             LBLᵀVTO ⟨ T ⟩
148             RCL 40
149             RCL 41
150             +
151             SQRT
152             RCL 40
153             SQRT
154             −
155             RCL 39
156             *
157             RCL 37
158             +
159             STO 37
160             RTN
161             END
```

PROGRAM PWIDTH

```
01          LBL⊤PWIDTH
02          FS? 10
03          GTO 40
04          CF 02
* * * CALCULATE Cbdj_avg, n ⟨ T ⟩ * * *
05          RCL 05
06          STO 39      ; FCn
07          RCL 06
08          STO 40      ; MJn
09          RCL 38      ; Vdd
10          .9
11          *
12          STO 41
13          RCL 07
14          STO 75      ; PBn ⟨ T ⟩
15          XEQ⊤CBDJ
16          RCL 37
17          STO 43
18          RCL 38
19          .1
20          *
21          STO 41
22          XEQ⊤CBDJ
23          RCL 37
24          RCL 43
25          +
26          2
27          /
28          STO 44
* * * CALCULATE Cbdj_avg, p ⟨ T ⟩ * * *
29          RCL 22
30          STO 39      ; FCp
31          RCL 23
32          STO 40      ; MJp
33          RCL 38      ; Vdd
34          .9
35          *
36          STO 41
37          RCL 24
38          STO 75      ; PBp⟨ T ⟩
39          XEQ⊤CBDJ
40          RCL 37
41          STO 43
42          RCL 38
43          .1
44          *
45          STO 41
46          XEQ⊤CBDJ
```

```
47          RCL 37
48          RCL 43
49          +
50          2
51          /
52          STO 45
* * * CALCULATE Cbds_avg, n ⟨ T ⟩ * * *
53          RCL 05
54          STO 39      ; FC_n
55          RCL 08
56          STO 40      ; MJSW_n
57          RCL 38      ; Vdd
58          .9
59          *
60          STO 41
61          RCL 07
62          STO 75      ; PB_n ⟨ T ⟩
63          XEQ^TCBDJ
64          RCL 37
65          STO 43
66          RCL 38
67          .1
68          *
69          STO 41
70          XEQ^TCBDJ
71          RCL 37
72          RCL 43
73          +
74          2
75          /
76          STO 46
* * * CALCULATE Cbds_avg, p ⟨ T ⟩ * * *
77          RCL 22
78          STO 39      ; FC_p
79          RCL 25
80          STO 40      ; MJSW_p
81          RCL 38      ; Vdd
82          .9
83          *
84          STO 41
85          RCL 24
86          STO 75      ; PB_p ⟨ T ⟩
87          XEQ^TCBDJ
88          RCL 37
89          STO 43
90          RCL 38
            .1
            *
            STO 41
94          XEQ^TCBDJ
```

```
95          RCL 37
96          RCL 43
97          +
98          2
99          /
100         STO 47
* * * CALCULATE C Σ gate * * *
                load
                devices
101         LBL 00
102         ᵀDO YOU KNOW CΣ L
103         ᵀ⊢OAD DEV ?
104         AVIEW
105         PSE
106         SF 27
107         ᵀYES _ _ _ _ _ _ NO
108         PROMPT
109         LBL E
110         CF 27
111         SF 10
112         ᵀLOADCAP
113         GETSUB
114         SF 03
115         XEQᵀLOADCAP
116         LBL 40
117         CF 10
118         ᵀLOADCAP
119         PCLPS
120         GTO 00
121         LBL A
122         CF 27
123         FC? 02
124         GTO 03
125         LBL 04
* * * ENTER MASK GEOMETRY DATA * * *
126         ᵀDESIRED RISE TI
127         ᵀ⊢ME ⟨ SEC ⟩ = ?
128         PROMPT
129         STO 49
130         ᵀROUTING CAP ⟨ FAR
131         ᵀ⊢AD ⟩ = ?
132         PROMPT
133         STO 50
134         ᵀP DRAIN MASK LN
135         ᵀ⊢GTH ⟨ MTR ⟩ ?
136         PROMPT
137         STO 51
138         ᵀN DRAIN MASK LN
139         ᵀ⊢GTH ⟨ MTR ⟩ ?
140         PROMPT
141         STO 52
```

```
142         ⊤NO. P DIFFUSION
143         ⊤⊢_CONTACTS
144         PROMPT
145         STO 53
146         ⊤PDIFCTC MASK LN
147         ⊤⊢GTH ⟨ MTR ⟩ ?
148         PROMPT
149         STO 54
150         ⊤PDIFCTC MASK WI
151         ⊤⊢DTH ⟨ MTR ⟩ ?
152         PROMPT
153         STO 55
154         NO. N DIFFUSION
155         ⊤⊢_CONTACTS
156         PROMPT
157         STO 56
158         ⊤NDIFCTC MASK LN
159         ⊤⊢GTH ⟨ MTR ⟩ ?
160         PROMPT
161         STO 57
162         ⊤NDIFCTC MASK WI
163         ⊤⊢DTH ⟨ MTR ⟩
164         PROMPT
165         STO 58
* * * CALCULATE "C" * * *
166         RCL 51
167         RCL 19
168         2
169         *
170         +
171         RCL 20
172         *
173         RCL 45
174         *
175         RCL 21
176         2
177         *
178         RCL 47
179         *
180         +
181         RCL 52
182         RCL 02
183         2
184         *
185         +
186         RCL 36
187         *
188         RCL 03
189         *
190         RCL 44
191         *
```

```
192             +
193             RCL 36
194             2
195             *
196             RCL 04
197             *
198             RCL 46
199             *
200             +
201             STO 59
* * * CALCULATE "D" * * *
202             RCL 51
203             RCL 19
204             2
205             *
206             +
207             RCL 20
208             *
209             RCL 45
210             *
211             RCL 21
212             2
213             *
214             RCL 47
215             *
216             +
217             RCL 19
218             *
219             2
220             *
221             STO 60
222             RCL 52
223             RCL 02
224             2
225             *
226             +
227             RCL 03
228             *
229             RCL 44
230             *
231             RCL 04
232             2
233             *
234             RCL 46
235             *
236             +
237             RCL 02
238             *
239             2
240             *
241             ST+ 60
```

```
242         RCL 51
243         RCL 19
244         2
245         *
246         +
247         RCL 53
248         RCL 54
249         *
250         +
251         2
252         *
253         RCL 21
254         *
255         RCL 47
256         *
257         ST+ 60
258         RCL 52
259         RCL 02
260         2
261         *
262         +
263         RCL 56
264         RCL 57
265         *
266         +
267         2
268         *
269         RCL 04
270         *
271         RCL 46
272         *
273         ST+ 60
274         RCL 55
275         RCL 19
276         2
277         *
278         +
279         RCL 20
280         *
281         RCL 45
282         *
283         RCL 55
284         RCL 28
285         *
286         +
287         RCL 54
288         *
289         RCL 53
290         *
291         ST+ 60
292         RCL 58
```

```
293         RCL 02
294         2
295         *
296         +
297         RCL 03
298         *
299         RCL 44
300         *
301         RCL 58
302         RCL 11
303         *
304         +
305         RCL 57
306         *
307         RCL 56
308         *
309         ST+ 60
* * * CALCULATE "F" * * *
310         RCL 42
311         2
312         RCL 19
313         *
314         -
315         2
316         *
317         RCL 49
318         /
319         RCL 17
320         /
321         RCL 38
322         RCL 35
323         -
324         /
325         STO 61
* * * CALCULATE "G" * * *
326         RCL 35
327         RCL 38
328         .1
329         *
330         -
331         RCL 38
332         RCL 35
333         -
334         /
335         STO 62
* * * CALCULATE "H" * * *
336         RCL 38
337         19
338         *
339         RCL 35
340         20
```

```
341         *
342         −
343         RCL 38
344         /
345         LN
346         2
347         /
348         STO 63
* * * CALCULATE Wp AND Wn * * *
349         RCL 62
350         RCL 63
351         +
352         RCL 59
353         *
354         RCL 61
355         *
356         1
357         X ⟨ ⟩ Y
358         −
359         1/X
360         RCL 61
361         *
362         RCL 62
363         RCL 63
364         +
365         *
366         RCL 48
367         RCL 60
368         +
369         RCL 50
370         +
371         *
372         STO 64
373         RCL 36
374         *
375         STO 65
* * * CALCULATE OUTPUT * * *
* * * PARAMETERS, OUTPUT * * *
* * * RESULTS, & RETURN TO * * *
* * * TOP-LVL * * *

* * * CALCULATE ADp = ASp * * *
376         RCL 64
377         RCL 19
378         2
379         *
380         +
381         RCL 51
382         RCL 19
383         2
```

```
384             *
385             +
386             *
387             RCL 55
388             RCL 19
389             2
390             *
391             +
392             RCL 53
393             *
394             RCL 54
395             *
396             +
397             STO 67
*** CALCULATE ADn = ASn ***
398             RCL 65
399             RCL 02
400             2
401             *
402             +
403             RCL 52
404             RCL 02
405             2
406             *
407             +
408             *
409             RCL 58
410             RCL 02
411             2
412             *
413             +
414             RCL 56
415             *
416             RCL 57
417             *
418             +
419             STO 70
*** CALCULATE PDp = PSp ***
420             RCL 51
421             RCL 19
422             2
423             *
424             +
425             RCL 53
426             RCL 54
427             *
428             +
429             2
430             *
431             RCL 64
432             RCL 19
```

```
433         2
434         *
435         +
436         2
437         *
438         +
439         STO 68
* * * CALCULATE PD_n = PS_n * * *
440         RCL 52
441         RCL 02
442         2
443         *
444         +
445         RCL 56
446         RCL 57
447         *
448         +
449         2
450         *
451         RCL 65
452         RCL 02
453         2
454         *
455         +
456         2
457         *
458         +
459         STO 71
* * * CALCULATE RD_p = RS_p * * *
460         RCL 55
461         RCL 19
462         2
463         *
464         +
465         1/X
466         RCL 54
467         *
468         RCL 53
469         /
470         STO 37
471         RCL 64
472         RCL 19
473         2
474         *
475         +
476         1/X
477         RCL 51
478         RCL 19
479         2
480         *
481         +
```

```
482             *
483             RCL 37
484             +
485             RCL 74
486             *
487             STO 69
* * * CALCULATE RD_n = RS_n * * *
488             RCL 58
489             RCL 02
490             2
491             *
492             +
493             1/X
494             RCL 57
495             *
496             RCL 56
497             /
498             STO 37
499             RCL 65
500             RCL 02
501             2
502             *
503             +
504             1/X
505             RCL 52
506             RCL 02
507             2
508             *
509             +
510             *
511             RCL 37
512             +
513             RCL 73
514             *
515             STO 72
* * * OUTPUT SPICE MOSFET PARAMETERS * * *
516             ᵀL ⟨ P ⟩ =
517             ARCL 42
518             ᵀ⊢⟨ MTR ⟩
519             PROMPT
520             ᵀW ⟨ P ⟩ =
521             ARCL 64
522             ᵀ⊢⟨ MTR ⟩
523             PROMPT
524             ᵀAD ⟨ P ⟩ =
525             ARCL 67
526             ᵀ⊢⟨ SQ. MTR ⟩
527             PROMPT
528             ᵀPD ⟨ P ⟩ =
529             ARCL 68
530             ᵀ⊢⟨ MTR ⟩
```

```
531          PROMPT
532          ᵀRD ⟨ P ⟩ =
533          ARCL 69
534          ᵀ⊢⟨ OHM ⟩
535          PROMPT
536          ᵀL ⟨ N ⟩ =
537          ARCL 66
538          ᵀ⊢⟨ MTR ⟩
539          PROMPT
540          ᵀW ⟨ N ⟩ =
541          ARCL 65
542          ᵀ⊢⟨ MTR ⟩
543          PROMPT
544          ᵀAD ⟨ N ⟩ =
545          ARCL 70
546          ᵀ⊢⟨ SQ. MTR ⟩
547          PROMPT
548          ᵀPD ⟨ N ⟩ =
549          ARCL 71
550          ᵀ⊢⟨ MTR ⟩
551          PROMPT
552          ᵀRD ⟨ N ⟩ =
553          ARCL 72
554          ᵀ⊢⟨ OHM ⟩
555          PROMPT
556          ᵀTOP-LVL
557          GETP
* * * Cbdj ⟨ T, Vbd ⟩, Cbds ⟨ T, Vbd ⟩ SUBROUTINE * * *
558          LBLᵀCBDJ
559          RCL 41
560          RCL 40
561          *
562          RCL 75
563          /
564          RCL 40
565          1
566          +
567          RCL 39
568          *
569          1
570          X ⟨ ⟩ Y
571          −
572          +
573          1
574          RCL 39
575          −
576          1
577          RCL 40
578          +
579          CHS
580          Y ↗ X
```

```
581         *
582         STO 37
583         RTN
584         LBL 03
585         ⊤C Σ LOAD DEV ⟨ FARA
586         ⊤⊢D ⟩ = ?
587         PROMPT
588         STO 48
589         GTO 04
590         END
```

PROGRAM LOADCAP

```
01          LBL"LOADCAP
02          SF 02
03          "LOADCAP
04          AVIEW
05          PSE
06          0
07          STO 37
08          LBL 05
09          "LD CHNL MASK LN
10          "⊢GTH ⟨ MTR ⟩
11          PROMPT
12          STO 39
13          "LD CHNL MASK WI
14          "DTH ⟨ MTR ⟩
15          PROMPT
16          STO 40
17          "DEVICE TYPE?
18          AVIEW
19          PSE
20          SF 27
21          "N _ _ _ _ _ _ _ _ _ _ P
22          PROMPT
    * * * CALCULATE WITH N DATA * * *
23          LBL A
24          CF 27
25          RCL 39
26          2
27          RCL 02
28          *
29          −
30          8.854187818  E-12
31          *
32          RCL 12
33          *
34          RCL 15
35          /
36          RCL 40
37          *
38          RCL 13
39          RCL 40
40          *
41          +
42          RCL 14
43          RCL 40
44          *
45          +
46          ST+ 37
47          "ANOTHER DEVICE?
```

```
48          AVIEW
49          PSE
50          SF 27
51          ┬    Y   N
52          PROMPT
53          LBL B
54          CF 27
55          GTO 05
56          LBL D
57          CF 27
58          GTO 06
    *** CALCULATE WITH P DATA ***
59          LBL E
60          CF 27
61          RCL 39
62          2
63          RCL 19
64          *
65          −
66          8.854187818  E-12
67          *
68          RCL 29
69          *
70          RCL 32
71          /
72          RCL 40
73          *
74          RCL 30
75          RCL 40
76          *
77          +
78          RCL 31
79          RCL 40
80          *
81          +
82          ST + 37
83          ┬ANOTHER DEVICE?
84          AVIEW
85          PSE
86          SF 27
87          ┬    Y   N
88          PROMPT
89          LBL 06
90          RCL 37
91          STO 48
92          FC? 03
93          GTO 07
94          RTN
95          LBL 07
96          ┬C Σ LOAD DEV=
97          ARCL 48
```

```
98          ⊤⊢( FARAD )
99          PROMPT
100         FS? 09
101         XEQ⊤TOP-LVL
102         XEQ⊤PWIDTH
102         END
```

APPENDIX B

STATIC CMOS SUPER BUFFER PSPICE DECK

This appendix contains the actual PSpice deck used for the simulation of the super buffer designed in Chapter 3. The worst-case minimum current MOSIS SPICE fabrication parameters are included in the deck.

```
SUPER BUFFER
M1 3 2 1 1 M1 L=3.0Um W=106.1Um AD=507.3P PD=246.0Um
M2 3 2 0 0 M2 L=3.0Um W=41.01Um AD=191.4P PD=102.6Um
M3 4 3 1 1 M3 L=3.0Um W=134.3Um AD=660.9P PD=314.4Um
M4 4 3 0 0 M4 L=3.0Um W=51.93Um AD=231.2P PD=124.4Um
C1 4 0 511.2FF
VDD 1 0 4.5
VIN 2 0
+PULSE 0 4.5 6Ns 0.01Ns 0.01Ns 6Ns 12Ns
.TEMP 85.0
.TRAN/OP 1.0NS 42NS 0NS 0.1NS
.PROBE
.PLOT TRAN V(2,0) V(4,0)
.PRINT TRAN V(2,0) V(4,0)
.OPTIONS NODE LIST OPTS ACCT LIMPTS=2000 NUMDGT=8 ITL5=0 ITL4=40
.WIDTH OUT = 80
.OP
*                                                                    *
*                                                                    *
* MOSIS CMOS WORST MINIMUM CURRENT CASE SPICE PARAMETERS             *
*                                                                    *
*                                                                    *
.MODEL M1 PMOS
+RD        =      32.76
+LEVEL     =      2.000
+VTO       =     -1.000
+KP        =   1.26D-05
+GAMMA     =      0.700
+PH        =      0.600
+LAMBDA    =   4.70D-02
+CGSO      =   4.00D-10
+CGDO      =   4.00D-10
+RSH       =     70.000
+CJ        =   3.50D-04
```

```
+MJ        =      0.500
+CJSW      =   2.00D-10
+MJSW      =      0.330
+TOX       =   5.50D-08
+NSUB      =  1.12D+14
+NSS       =  0.00D+00
+NFS       =  8.80D+11
+TPG       =     -1.000
+XJ        =   4.00D-07
+LD        =   4.80D-07
+UO        =    200.000
+UCRIT     =  1.60D+04
+UEXP      =      0.150
+VMAX      =  1.00D+05
+NEFF      =      0.010
+DELTA     =      1.900
*
.MODEL M2 NMOS
+RD        =      29.73
+LEVEL     =      2.000
+VTO       =      1.000
+KP        =   3.77D-05
+GAMMA     =      1.500
+PHI       =      0.600
+LAMBDA    =   1.60D-02
+CGSO      =   5.20D-10
+CGDO      =   5.20D-10
+RSH       =     30.000
+CJ        =   2.20D-04
+MJ        =      0.500
+CJSW      =   3.00D-10
+MJSW      =      0.330
+TOX       =   5.50D-08
+NSUB      =  1.00D+16
+NSS       =  0.00D+00
+NFS       =  1.20D+12
+TPG       =      1.000
+XJ        =   6.00D-07
+LD        =   3.20D-07
+UO        =    600.000
+UCRIT     =  9.99D+05
+UEXP      =      0.001
+VMAX      =  1.00D+05
+NEFF      =      0.010
+DELTA     =      1.200
*
.MODEL M3 PMOS
+RD        =      22.16
+LEVEL     =      2.000
+VTO       =     -1.000
+KP        =   1.26D-05
```

```
+GAMMA   =      0.700
+PHI     =      0.600
+LAMBDA  =   4.70D-02
+CGSO    =   4.00D-10
+CGDO    =   4.00D-10
+RSH     =     70.000
+CJ      =   3.50D-04
+MJ      =      0.500
+CJSW    =   2.00D-10
+MJSW    =      0.330
+TOX     =   5.50D-08
+NSUB    =   1.12D+14
+NSS     =   0.00D+00
+NFS     =   8.80D+11
+TPG     =     -1.000
+XJ      =   4.00D-07
+LD      =   4.80D-07
+UO      =     200.000
+UCRIT   =   1.60D+04
+UEXP    =      0.150
+VMAX    =   1.00D+05
+NEFF    =      0.010
+DELTA   =      1.900
*
.MODEL M4 NMOS
+RD      =     29.19
+LEVEL   =      2.000
+VTO     =      1.000
+KP      =   3.77D-05
+GAMMA   =      1.500
+PHI     =      0.600
+LAMBDA  =   1.60D-02
+CGSO    =   5.20D-10
+CGDO    =   5.20D-10
+RSH     =     30.000
+CJ      =   2.20D-04
+MJ      =      0.500
+CJSW    =   3.00D-10
+MJSW    =      0.330
+TOX     =   5.50D-08
+NSUB    =   1.00D+16
+NSS     =   0.00D+00
+NFS     =   1.20D+12
+TPG     =      1.000
+XJ      =   6.00D-07
+LD      =   3.20D-07
+UO      =     600.000
+UCRIT   =   9.99D+05
+UEXP    =      0.001
+VMAX    =   1.00D+05
+NEFF    =      0.010
```

```
+DELTA     =        1.200
*
*
*       SPICE node  Signal name
*
*             0         GND
*             1         VDD
*             2         VIN
*             3         OUT
*             4         INTERMEDIATE
.END
```

LIST OF REFERENCES

Baumstarck, J. E., *SCMOS Silicon Compiler Organelle Design and Insertion*, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, December 1987.

Electronics Research Laboratory, College of Engineering, University of California at Berkeley, Memo No. UCB/ERL M80/7, *The Simulationn of MOS Integrated Circuits Using SPICE2*, by A. Vlademirescu and S. Liu, February 1980.

Hodges, D. A. and Jackson, H. G., *Analysis and Design of Digital Integrated Circuits*, McGraw-Hill Book Company, 1983.

Information Sciences Institute, University of Southern California, The MOSIS Project ISI/TM-84-128, *The MOSIS System (What it is and How to Use it)*, March 1984.

Malagon, E. G., *Technology Upgrade of a Silicon Compiler*, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, June 1987.

Massachusetts Institute of Technology 1982 Conference on Advanced Research in VLSI, *Generating Custom High Performance VLSI Designs from Succinct Algorithmic Descriptions*, by J. M. Siskind, J. R. Southard, and K. W. Crouch, 25 January 1982.

McCarthy, O. J., *MOS Devices and Circuit Design*, John Wiley & Sons, 1982.

Mead, C. and Conway, L., *Introduction to VLSI Systems*, Addison-Wesley Publishing Company, 1980.

MicroSim Corp., *PSpice*, 1987.

Mullarky, A. J., *CMOS Cell Library for a Silicon Compiler*, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, March 1987.

Pollack, S., Erickson, B., and Mazor, S., "Silicon Compilers Ease Complex VLSI Design," *Computer Design*, pp. 79-82, 15 September 1986.

Sedra, A. S. and Smith, K. C., *Micro-Electronic Circuits*, Holt, Rinehart and Winston, 1982.

Signetics Corp., *High-Speed CMOS Data Manual*, 1986.

Weste, N. H. E. and Eshraghian, K., *Principles of CMOS VLSI Design: A Systems Perspective*, Addison-Wesley Publishing Company, 1985.

Wyatt Jr., J. L., "*The Practical Engineer's No-Nonsense Guide to On-Chip Signal Delay Calculations*," VLSI Memo No. 87-381, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1987 (private communication).

BIBLIOGRAPHY

Annaratone, M., ***Digital CMOS Circuit Design***, Klewer Academic Publishers, 1986.

Baumstarck, J. E., ***SCMOS Silicon Compiler Organelle Design and Insertion***, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, December 1987.

Boylestead, R. and Nashelsky, L., ***Electronic Devices and Circuit Theory***, 3rd ed., Prentice Hall, Inc., 1972.

Carlson, D. J., ***Application of a Silicon Compiler to VLSI Design of Digital Pipelined Multipliers***, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, June 1984.

Electronics Research Laboratory, College of Engineering, University of California at Berkeley, Memo No. UCB/ERL M80/7, ***The Simulation of MOS*** Integrated Circuits Using SPICE2, by A. Vlademirescu and S. Liu, February 1980.

Hodges, D. A. and Jackson, H. G., ***Analysis and Design of Digital Integrated*** Circuits, McGraw-Hill Book Company, 1983.

Information Sciences Institute, University of Southern California, The MOSIS Project ISI/TM-84-128, ***The MOSIS System (What it is and How to Use it)***, March 1984.

Kim, J., "***PATH: A Simulation-Based Transistor Sizer***," VLSI Memo No. 87-412, Massachusetts Institute of Technology, Cambridge, Massachusetts, September 1987 (private communication).

Lincoln Laboratory, Massachusetts Institute of Technology, Project Report RVLSI-3, ***An Introduction to MacPITTS***, by J. R. Southard, 10 February 1983.

Lincoln Laboratory, Massachusetts Institute of Technology, Project Report RVLSI-5. ***L5 User's Guide***, by K. W. Crouch, 7 March 1984.

Malagon, E. G., ***Technology Upgrade of a Silicon Compiler***, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, June 1987.

Malagon-Fajar, M. A., ***Silicon Compilation Using a Lisp-Based Layout Language***. M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, June 1986.

Massachusetts Institute of Technology 1982 Conference on Advanced Research in VLSI. ***Generating Custom High Performance VLSI Designs from Succinct Algorithmic Descriptions***, by J. M. Siskind, J. R. Southard, and K. W. Crouch, 25 January 1982.

McCarthy, O. J., ***MOS Devices and Circuit Design***, John Wiley & Sons, 1982.

Mead, C. and Conway, L., ***Introduction to VLSI Systems***, Addison-Wesley Publishing Company, 1980.

Meindl, J. D., "Chips for Advanced Computing," ***Scientific American***, v. 257, pp. 78-88, October 1987.

MicroSim Corp., ***PSpice***, 1987.

Motorola Inc., ***CMOS Integrated Circuits***, 1978.

Motorola Inc., ***CMOS/NMOS Special Functions Data***, 1986.

Mullarky, A. J., ***CMOS Cell Library for a Silicon Compiler***, M.S.E.E. Thesis, Naval Postgraduate School, Monterey, California, March 1987.

Muller, R.S. and Kamins T. I., ***Device Electronics for Integrated Circuits***, John Wiley & Sons, 1977.

Mulvey, J., ***Semiconductor Device Measurements***, TEKTRONIX, Inc., Beaverton, Oregon, 1968.

National Semiconductor Corp., ***CMOS Integrated Circuits***, 1975.

National Semiconductor Corp., ***CMOS Databook***, 1981.

Nilsson, J. W., ***Electric Circuits***, Addison-Wesley Publishing Company, 1984.

Pollack, S., Erickson, B., and Mazor, S., "Silicon Compilers Ease Complex VLSI Design," ***Computer Design***, pp. 79-82, 15 September 1986.

RCA Corp. Solid State, ***CMOS Integrated Circuits Databook***, 1983.

Sedra, A. S. and Smith, K. C., ***Micro-Electronic Circuits***, Holt, Rinehart and Winston. 1982.

Signetics Corp., ***High-Speed CMOS Data Manual***, 1986.

Texas Instruments Inc., ***High-Speed CMOS Logic Data Book (Silicon-Gate Complementary MOS)***, 1984.

University of California at Berkeley Report No. UCB/CSD 86/272, ***1986 VLSI Tools: Still More Works by the Original Artists***, edited by W. S. Scott, R. N. Mayo, G. Hamachi, and J. K. Ousterhoot, December 1985.

Weste, N. H. E. and Eshraghian, K., ***Principles of CMOS VLSI Design: A Systems Perspective***, Addison-Wesley Publishing Company, 1985.

Wilensky R., ***LISPcraft***, W. W. Norton & Company, 1984.

Wyatt Jr., J. L., "***The Practical Engineer's No-Nonsense Guide to On-Chip Signal Delay Calculations***," VLSI Memo No. 87-381, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1987 (private communication).

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Dr. D. E. Kirk, Code 62Ki Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	2
5.	Dr. H. H. Loomis, Jr., Code 62Lm Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	2
6.	Dr. C. Yang, Code 62Ya Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
7.	Mr. D. Schaeffer, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
8.	Dr. B. Wei Department of Electrical Engineering San Jose State University San Jose, CA 95192-0080	1

9. Mr. P. Blankenship — 1
Massachusetts Institute of Technology
Lincoln Laboratory
P. O. Box 73
Lexington, MA 02173-0073

10. Mr. J. O'Leary — 1
Massachusetts Institute of Technology
Lincoln Laboratory
P. O. Box 73
Lexington, MA 02173-0073

11. Dr. T. Bestul — 1
Naval Research Laboratories
Code 7590
Washington, D. C. 20375

12. Mr. A. DeGroot — 1
Lawrence Livermore National Laboratory
P. O. Box 808
Livermore, CA 94550

13. Lt Col A. Ross, USAF — 1
Naval Research Laboratory, Code 9110
4555 Overlook Ave., S. W.
Washington, D. C. 20375

14. CDR D. Southworth, USN — 1
Office of Naval Technology, Code ONT227
800 N. Quincy (BT#1)
Arlington, VA 22217-5000

15. Mr. J. Hall — 1
Office of Naval Technology, Code ONT20P4
800 N. Quincy (BT#1)
Arlington, VA 22217-5000

16. Deputy Commanding General — 1
Marine Corps Research, Development, and Acquisition Command
Code C2IS
Quantico, VA 22134

17. Commandant of the Marine Corps — 1
Headquarters, United States Marine Corps
Code INTS
Washington, D. C. 20380-0001

18. Mrs. B. Steele — 1
Digital Equipment Corporation
MK02-2/F10
Continental Boulevard
Merrimack, NH 03054

19. Mr. J. McCracken — 1
United States Department of Labor
Room S 1310
Frances Perkins Building
200 Constitution Ave. N. W.
Washington, D. C. 20210

20. LtCol W. H. Keller, III, USMC — 1
110 Owings Hill Court
Owings, MD 20736

21. Mrs. Robert Limes — 1
411 Evergreen Road
Pacific Grove, CA 93950

22. Col M. Sanger, USA — 1
1803 Rupert Street
MacLean, VA 22101

23. Capt G. Steele, USMC — 3
Marine Corps Research, Development, and Acquisition Command
Code C2IS
Quantico, VA 22134